

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

MIN TAO* AND XIAOMING YUAN†

December 31 2009

Abstract. Many applications arising in a variety of fields can be well illustrated by the task of recovering the low-rank and sparse components of a given matrix. Recently, it is discovered that this 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. In the literature, it is conventionally assumed that all entries of the matrix to be recovered are exactly known (via observation).

To capture even more applications, this paper studies the recovery task in more general settings: only a fraction of entries of the matrix can be observed and the observation is corrupted by both impulsive and Gaussian noise. The resulted 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 implementable numerical algorithms are developed in the spirits of the well-known alternating direction method and the parallel splitting augmented Lagrangian method. Some preliminary numerical experiments verify that these augmented-Lagrangian-based algorithms are easily-implementable and surprisingly-efficient for tackling the new recovery model.

Key words. Matrix recovery, principle component analysis, sparse, low rank, alternating direction method, augmented Lagrangian method.

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

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, system identification in engineering, [19, 20, 21, 38, 44, 46, 53]. 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 NP hard, can be accurately accomplished in the probabilistic sense via solving the following nuclear-norm- and l_1 -norm-involved convex relaxation problem:

$$\begin{aligned} \min_{A,E} \quad & \|A\|_* + \tau \|E\|_1 \\ \text{s.t.} \quad & A + E = C, \end{aligned} \tag{1.1}$$

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

*DEPARTMENT OF MATHEMATICS, NANJING UNIVERSITY, NANJING, JIANGSU, CHINA (TAOM@NJUPT.EDU.CN).

†DEPARTMENT OF MATHEMATICS, HONG KONG BAPTIST UNIVERSITY, HONG KONG, CHINA (XYUAN@HKBU.EDU.HK). THIS AUTHOR WAS SUPPORTED BY THE RGC GRANT 203009 AND THE NSFC GRANT 10701055.

sparse component of C ; and $\tau > 0$ is a constant balancing the low-rank and sparsity. This model has also been highlighted in [54] in the context of the so-called robust principle 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 principle components and E is the corrupted data matrix which is sparse yet its entries can be arbitrary in magnitude. As analyzed in [54], the model (1.1) is capable of recovering the principle 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 PCA which can only complete exact recovery from mildly (but could be densely) corrupted observation data, e.g. [33].

In [54], theoretical conditions to ensure the perfect recovery in the probabilistic sense via the convex programming (1.1) have been profoundly analyzed, with the defaulted assumption that the dimensionality of data (i.e., m) is sufficiently high. This benefit brought by the high dimensionality was dubbed in [13] 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 [47] and SDPT3 [51]. 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 contributions on numerical approaches to (1.1), e.g., the iterative thresholding algorithm [54]; the accelerated proximal gradient algorithm in [40], and the alternating direction method proposed almost simultaneously and independently in [39, 59].

As pointed out at the end in [54], in any real world application, we need to consider the model (1.1) under more practical circumstance. 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. [5, 6, 7, 8, 34, 45], for the justification of considering incomplete observation. In particular, let Ω be a subset of the index set of entries $\{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$. 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 non-zero entries of E . Adhere to the notation of [5, 7], we also summarize the incomplete observation information by the operator $P_\Omega : \mathcal{R}^{m \times n} \rightarrow \mathcal{R}^{m \times n}$, which is the orthogonal projection onto the span of matrices vanishing outside of Ω 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 principle 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 $\|P_\Omega(F)\|_F \leq \delta$, 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):

$$\begin{aligned} \min_{A,E} \quad & \|A\|_* + \tau\|E\|_1 \\ \text{s.t.} \quad & \|P_\Omega(C - A - E)\|_F \leq \delta. \end{aligned} \quad (1.2)$$

Note that the model (1.2) includes many special yet self-interested cases which capture diversified applications in many fields. For instance, the Gaussian-noisy low-rank and sparse recovery with complete observation (or RPCA with Gaussian noise as in (133) of [54]):

$$\begin{aligned} \min_{A,E} \quad & \|A\|_* + \tau\|E\|_1 \\ \text{s.t.} \quad & \|C - A - E\|_F \leq \delta; \end{aligned} \quad (1.3)$$

the Gaussian-noiseless low-rank and sparse recovery with incomplete observation (RPCA without Gaussian noise as in ((135) of [54])):

$$\begin{aligned} \min_{A,E} \quad & \|A\|_* + \tau\|E\|_1 \\ \text{s.t.} \quad & P_\Omega(A + E) = P_\Omega(C); \end{aligned} \quad (1.4)$$

the Gaussian-noiseless low-rank and sparse recovery with complete observation (1.1); the noiseless and noisy matrix completion problem where $\tau = 0$ and $E = 0$ in [7, 35].

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); and 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, depending on the fashion of splitting the augmented Lagrangian function, we shall develop the alternating splitting augmented Lagrangian method (ASALM) and its variant (VASALM), and the parallel splitting augmented Lagrangian method (PSALM) 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 ASALM which admits nicer convergence. As the concurrence of ASALM, the PSALM, which differs from ASALM in that it splits the augmented Lagrangian function in the parallel manner, is also derived in Section 7.. In Section 8., 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 9., 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 10., 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 concerning 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 $\mathbf{T} \in \mathcal{R}^{m \times n}$, the solution of the following problem*

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

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

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

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

Proof. See, e.g. [4, 11, 56]. \square

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

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

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

$$\mathcal{D}_\mu(X) := U \operatorname{diag}(S_\mu(\Sigma)) V^T, \quad (2.2)$$

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 :

$$X = U \Sigma V^T, \quad \text{and} \quad \Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \dots, \sigma_r).$$

Proof. See, e.g. [5, 32, 41]. \square

The following result is useful in the convergence proof of the proposing algorithms.

LEMMA 2.3. *Let \mathcal{H} be a real Hilbert space endowed with an inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\|\cdot\|$; $y \in \partial\|x\|$ where $\partial(\cdot)$ denote the subgradient operator. Then $\|y\|^* = 1$, if $x \neq 0$, and $\|y\|^* \leq 1$, if $x = 0$, where $\|\cdot\|^*$ is the dual norm of $\|\cdot\|$.*

Proof. See, e.g. [7, 40]. \square

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:

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

Let Λ be the Lagrange multiplier associated with the linear constraint in (3.1) (or the variable of the dual problem of (3.1)). Then, it is obvious that the optimal 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:

$$\begin{cases} 0 \in \partial(\|A^*\|_*) - \Lambda^*, \\ 0 \in \tau\partial(\|E^*\|_1) - \Lambda^*, \\ \langle Z' - Z^*, -\Lambda^* \rangle \geq 0, \forall Z' \in \mathbf{B}, \\ A^* + E^* + Z^* - M = 0, \end{cases} \quad (3.2)$$

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 [17]).

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

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

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 Λ^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

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

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

3. Update β_{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 the literature, e.g. [2, 43]. In particular, it is easy to prove that the sequence of the Lagrange multiplier $\{\Lambda^k\}$ is F ej er monotone.

Hence, the convergence of 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:

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

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

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

Then, according to Lemma 2.3, the sequence $\{\Lambda^k\}$ is bounded since the dual norm of $\|\cdot\|_*$ is $\|\cdot\|_2$.

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 $\{\mathcal{L}_A(A^{k+1}, E^{k+1}, Z^{k+1}, \Lambda^k, \beta_k)\}$ 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, according to the third variational inequality (VI), we have

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

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)). \square

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 the optimal solution of (3.1).*

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

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

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 $\lim_{k \rightarrow \infty} \beta_k = +\infty$, we have

$$\lim_{k \rightarrow \infty} A^{k+1} + E^{k+1} + Z^{k+1} - M = 0. \quad (4.5)$$

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

5. The alternating splitting augmented Lagrangian method. As we emphasized previously, the direct application of 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 next 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 alternating splitting augmented Lagrangian method (ASALM) and its variant (VASALM) for solving (3.1) will be developed.

Recall that the ADM dates back to [23, 24, 25, 26] and is closely related to the Douglas-Rachford operator splitting method [12]; and it has attracted wide attention of many authors in various areas, see e.g. [3, 10, 14, 15, 22, 29, 31, 36, 49, 50]. In particular, some novel and attractive applications of ADM have been discovered very recently, e.g. the total-variation problem in Image Processing [16, 42, 49, 57], the covariance selection problem and semidefinite least square problem in Statistics [30, 59], the SDP [48, 55], 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, Λ^k) , the ASALM generates the new iterate $(A^{k+1}, E^{k+1}, Z^{k+1}, \Lambda^{k+1})$ via the following scheme:

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

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

$$\begin{cases} Z^{k+1} = \operatorname{argmin}_{Z \in \mathbf{B}} \frac{\beta}{2} \|Z + A^k + E^k - \frac{1}{\beta} \Lambda^k - M\|_F^2, \\ E^{k+1} = \operatorname{argmin}_{E \in \mathcal{R}^{m \times n}} \tau \|E\|_1 + \frac{\beta}{2} \|E + A^k + Z^{k+1} - \frac{1}{\beta} \Lambda^k - M\|_F^2, \\ A^{k+1} = \operatorname{agrmin}_{A \in \mathcal{R}^{m \times n}} \|A\|_* + \frac{\beta}{2} \|A + E^{k+1} + Z^{k+1} - \frac{1}{\beta} \Lambda^k - M\|_F^2, \\ \Lambda^{k+1} = \Lambda^k - \beta(A^{k+1} + E^{k+1} + Z^{k+1} - M). \end{cases} \quad (5.2)$$

As shown in (5.1) and (5.2), like the ADM method, the ASALM decomposes the minimization task in (4.2) into three separable task. 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 subproblems takes advantage of solutions of the first two subproblems in order to approximate the original augmented Lagrangian method 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 satisfactory numerical results of ASALM to be reported.

The main fact making the proposing ASALM easily implementable is that all the generated subproblems in (5.2) admit analytic 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:

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

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 $\mathcal{S}_{\tau/\beta}$ 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:

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

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 $(A^k, E^k, Z^k, \Lambda^k)$ 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 Λ^{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 $Z^k \rightarrow E^k \rightarrow A^k$ is that we actually allow partial SVD in the computation of A^k , as to be delineated in Section 9. Hence, to prevent the error resulted by the partial SVD from affecting the later solution, we do not follow the conventional alternating order: $A^k \rightarrow E^k \rightarrow Z^k$. We must also point out that if full SVD is executed in the A^k -involved subproblems, the convergence of the ASALM is valid no matter which alternating order among the variables $\{A^k, E^k, Z^k\}$ is used.

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

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

which is equivalent to

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

Recall the optimal condition (3.2). The characterization (5.5) immediately shows that the distant 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:

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

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 β 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 β 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
$$0 \in \partial(\|A^{k+1}\|_*) + \beta_k(A^{k+1} + E^{k+1} + Z^{k+1} - \frac{1}{\beta_k}\Lambda^k - M);$$

which equivalently states that

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

Then, according to Lemma 2.3, both 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.

Since 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 $\{\mathcal{L}_A(A^{k+1}, E^{k+1}, Z^{k+1}, \Lambda^k, \beta^k)\}$ 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 $\{\Lambda^k\}$. \square

Without more conditions, we must point out that the convergence of the proposed ASALM is ambiguous. In fact, although the convergence of 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 to hold, e.g. the condition proposed in [39] (i.e., $\lim_{k \rightarrow \infty} \beta_k(A^{k+1} - A^k) = 0$ and $\lim_{k \rightarrow \infty} \beta_k(E^{k+1} - E^k) = 0$), it is easy to derive the convergence of ASALM. For succinctness, we omit the proof and refer to [39].

6. The 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 ASALM under mild assumptions on $\{\beta_k\}$ is still ambiguous. Hence, although empirically the numerical performance of ASALM is shown to be dominantly better (as to be reported), we are still desired to seek some variants of ASALM whose convergence is ensured without restrictive requirements on the penalty parameter. This desire inspires us to propose the following variant of ASALM (denoted by VASALM).

6.1. Algorithm. More specifically, let \mathbf{B} be defined in (3.1). Let $\beta > 0$; $\eta > 0$ and $(A^k, E^k, Z^k, \Lambda^k)$ 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:

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

Analogously, the generated subproblems of VASALM are all easy to handle in the sense that they all admit analytical 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 ASALM for easy implementation is preserved completely by VASALM. For succinctness, we omit the elaboration on solving the subproblems of VASALM, which is analogous to that in Section 5.1. Instead, we directly present the algorithm of the VASALM for solving (3.1).

Let $\eta > 2$ and $\beta > 0$. Let (A^k, E^k, Λ^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

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

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 Λ^{k+1} via

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

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 VASALM, the Z^{k+1} -related subproblem should precede the E^{k+1} - and A^{k+1} -related subproblems. This is the alternating characterize of 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 characterize of 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 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:

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

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. 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 $\tilde{A}^k := A^{k+1}$.

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:*

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

where

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

and

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

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*

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

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

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

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

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

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

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

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

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

By setting $Z' = Z^*$, $E' = E^*$, and $A' = A^*$ in (6.9) and adding the resulted inequalities up to (6.10), we obtain

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

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

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

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

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

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

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

Therefore, it follows from Lemma 6.1 that

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

which is the assertion of this lemma. \square

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

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

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

where

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

Proof. Recall that the new iterate $V^{k+1} = (E^{k+1}, A^{k+1}, \Lambda^{k+1})$ 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 following 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

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

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 VASALM.

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

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

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

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

which immediately implies that

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

i.e., the second assertion. \square

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*

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

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

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

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

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

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

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 $\lim_{k \rightarrow \infty} \|\Lambda^{k+1} - \tilde{\Lambda}^k\| = 0$.

Based on all these facts, we then have

$$\left\{ \begin{array}{ll} \lim_{k \rightarrow \infty} \langle Z' - Z^{k+1}, -\Lambda^{k+1} \rangle \geq 0, & \forall Z' \in \mathbf{B}; \\ \lim_{k \rightarrow \infty} \langle E' - E^{k+1}, \tau G_1 - \Lambda^{k+1} \rangle \geq 0, & \forall G_1 \in \partial \|E^{k+1}\|_1, \forall E' \in \mathcal{R}^{m \times n}; \\ \lim_{k \rightarrow \infty} \langle A' - A^{k+1}, G_2 - \Lambda^{k+1} \rangle \geq 0, & \forall G_2 \in \partial \|A^{k+1}\|_*, \forall A' \in \mathcal{R}^{m \times n}; \\ \lim_{k \rightarrow \infty} Z^{k+1} + E^{k+1} + A^{k+1} - M = 0. \end{array} \right. \quad (6.18)$$

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 Fejér monotone with respect to the set \mathcal{V}^* , see e.g. [1]. \square

Compared to ASALM, 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 ASALM under mild conditions on β (e.g. any positive constant required by VASALM) is 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:*

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

where Ω_i ($i = 1, 2, 3$) are given closed convex subsets in $\mathcal{R}^{m \times n}$; $\theta_i : \Omega_i \rightarrow \mathcal{R}$ ($i = 1, 2, 3$) are given proper convex functions (not necessarily smooth); $\mathcal{B}_i : \Omega_i \rightarrow \mathcal{R}^{m \times n}$ ($i = 1, 2, 3$) 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 senses, 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. The parallel splitting augmented Lagrangian method. Recall that the generated subproblems of the proposed ASALM can only be solved in the consecutive order, for solutions of all the preceding subproblems are required by the current subproblem. On the other hand, the VASALM differs from ASALM mainly in that two of its three subproblems at each iterations are eligible for parallel computation. By doing so, the theoretical pitfall of ASALM on convergence can be remedied under more relaxed conditions. Hence, a natural question arises: Is it possible to split the augmented Lagrangian in the completely parallel manner, i.e., all the generated

subproblems are eligible for parallel computation, and thus develop the parallel splitting augmented Lagrangian method (PSALM) for (3.1)? This section is to answer this question affirmatively. In fact, if we split the augmented Lagrangian function in the parallel fashion, the convergence can be derived easily provided that some correction steps are performed; and this treatment falls into exactly an existing method developed in the literature [28]. Note that another motivation for us to consider the ways of splitting the augmented Lagrangian function in the parallel manner is to take advantage of the circumstance that parallel computation's facilities are available.

The intuition for solving the generated subproblems in parallel results easily in the following iterative scheme:

$$\begin{cases} Z^{k+1} \in \arg \min_{Z \in \mathbf{B}} \mathcal{L}_{\mathcal{A}}(A^k, E^k, Z, \Lambda^k, \beta_k), \\ E^{k+1} \in \arg \min_{E \in \mathcal{R}^{m \times n}} \mathcal{L}_{\mathcal{A}}(A^k, E, Z^k, \Lambda^k, \beta_k), \\ A^{k+1} \in \arg \min_{A \in \mathcal{R}^{m \times n}} \mathcal{L}_{\mathcal{A}}(A, E^k, Z^k, \Lambda^k, \beta_k), \\ \Lambda^{k+1} = \Lambda^k - \beta_k(A^{k+1} + E^{k+1} + Z^{k+1} - M), \end{cases} \quad (7.1)$$

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

$$\begin{cases} Z^{k+1} = \operatorname{argmin}_{Z \in \mathbf{B}} \frac{\beta}{2} \|Z + A^k + E^k - \frac{1}{\beta} \Lambda^k - M\|_F^2, \\ E^{k+1} = \operatorname{argmin}_{E \in \mathcal{R}^{m \times n}} \tau \|E\|_1 + \frac{\beta}{2} \|E + A^k + Z^k - \frac{1}{\beta} \Lambda^k - M\|_F^2, \\ A^{k+1} = \operatorname{argmin}_{A \in \mathcal{R}^{m \times n}} \|A\|_* + \frac{\beta}{2} \|A + E^k + Z^k - \frac{1}{\beta} \Lambda^k - M\|_F^2, \\ \Lambda^{k+1} = \Lambda^k - \beta(A^{k+1} + E^{k+1} + Z^{k+1} - M). \end{cases} \quad (7.2)$$

Unfortunately, convergence of the direct parallel splitting scheme (7.2) seems still ambiguous. The loss of convergence, however, can be picked up by the method in [28] where an additional descent step is employed to correct the iterate generated by (7.2). Hence, when the method in [28] applied, we derive the implementable PSALM for (3.1). Let $\beta > 0$ and $\gamma \in (0, 2)$, for the given (A^k, E^k, Λ^k) , the algorithm of PSALM generates the new iterate via the following computation.

The k -th iteration of the PSALM for (3.1):

1. Compute \tilde{Z}^k via

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

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

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

3. Compute \tilde{A}^k via $\tilde{A}^k = \mathcal{D}_{1/\beta}(\frac{1}{\beta}\Lambda^k + M - Z^k - E^k)$.

4. Update $\tilde{\Lambda}^k$ via $\tilde{\Lambda}^k = \Lambda^k - \beta(\tilde{A}^k + \tilde{E}^k + \tilde{Z}^k - M)$.

5. Generate the new iterate by correcting $\tilde{W}^k := (\tilde{Z}^k, \tilde{E}^k, \tilde{A}^k, \tilde{\Lambda}^k)$:

$$W^{k+1} = W^k - \alpha_k(W^k - \tilde{W}^k).$$

where

$$\alpha_k = \gamma \frac{\varphi(W^k, \tilde{W}^k)}{\|W^k - \tilde{W}^k\|_J};$$

$$J = \begin{pmatrix} 2\beta I_m & \beta I_m & \beta I_m & 0 \\ \beta I_m & 2\beta I_m & \beta I_m & 0 \\ \beta I_m & \beta I_m & 2\beta I_m & 0 \\ 0 & 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix};$$

and

$$\varphi(W^k, \tilde{W}^k) := \|W^k - \tilde{W}^k\|_J^2 + 2\langle \Lambda^k - \tilde{\Lambda}^k, Z^k - \tilde{Z}^k + E^k - \tilde{E}^k + A^k - \tilde{A}^k \rangle.$$

Convergence of the proposed PSALM is easily derived according to [28]. Hence, omitted.

8. Extension. An alternative model to study (1.2) is the following nuclear-norm- and l_1 -norm-regularized least squares problem:

$$\min_{A, E} \|A\|_* + \tau \|E\|_1 + \frac{1}{2\mu} \|P_\Omega(C - A - E)\|_F^2, \quad (8.1)$$

which is an extended model of what the existing literature has addressed, e.g. [5, 41, 52]. In this section, we extend the proposed ASALM to solve (8.1) for the completeness.

By reformulating (8.1) into the following favorable form:

$$\begin{aligned} \min_{A, E, Z} \quad & \|A\|_* + \tau \|E\|_1 + \frac{1}{2\mu} \|P_\Omega(Z)\|_F^2 \\ \text{s.t.} \quad & A + E + Z = M, \end{aligned} \quad (8.2)$$

where $M = P_\Omega(C)$ as defined before, then the proposed ASALM is easily extended. In

fact, the iterative scheme of ASALM for (8.2) consists of the following subproblems:

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

Let $\beta > 0$ and (A^k, E^k, Λ^k) be given. Then, the ASALM for (8.2) 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 (8.1):

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

Remark. The proposed VASALM and PSALM can also be easily extended to solve (8.1). For succinctness, we skip the detail and only focus on the extension of ASALM for (8.1).

9. Numerical Experiments. In this section, we test the proposed ASALM, VASALM and PSALM 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 considerably 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. [5, 52]. As well-known, however, PROPACK is not able to compute automatically those singular values larger than the particular threshold, even 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, for ASALM and VASALM, 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 the variation of its rank is very small. Then, we switch to PROPACK to execute partial SVD. On the opposite, for PSALM, we always execute full.

In the following, **sr**, **spr** and **rr** represent the ratios of sample (observed) entries (i.e., $|\Omega|/mn$), the number of non-zero 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.

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

The implementation of these three methods were terminated whenever the following stopping criterion is satisfied:

$$\text{RelChg} := \frac{\|(A^{k+1}, E^{k+1}) - (A^k, E^k)\|_F}{\|(A^k, E^k)\|_F + 1} \leq 10^{-6}, \quad (9.1)$$

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 (9.1) 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 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. Ω , is determined at random. The support $\Gamma \subset \Omega$ of the impulsive noise E^* (sparse but large) are chosen uniformly at random, and the non-zero entries of E^* are i.i.d. uniformly in the interval $[-500, 500]$. We set $\tau = 1/\sqrt{n}$ in (1.2).

The SVD computation deserves further explanation. As we remarked before, PROPACK saves computation considerably 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 ASALM and VASALM, but not at all for PSALM. To illustrate this observation, for the case that $m = n = 500$, **rr** = **spr** = 0.05, **sr** = 0.8, we execute the full SVD (by an Mex File in C interface) to compute the exact rank of A^k (denoted by $\text{Rank}(A^k)$), and we record the variations of $\text{Rank}(A^k)$ for these three methods. The different variations are plotted in the left of the Figure 9.1.

As shown in Figure 9.1, the rank of A^k generated by PSALM changes radically according to iterations at the first stage; while the rank of A^k generated by ASALM and VASALM are much more stable—not sensitive to the iterations. We believe that the reason behind this difference is that the correction step of PSALM (Step 5 in

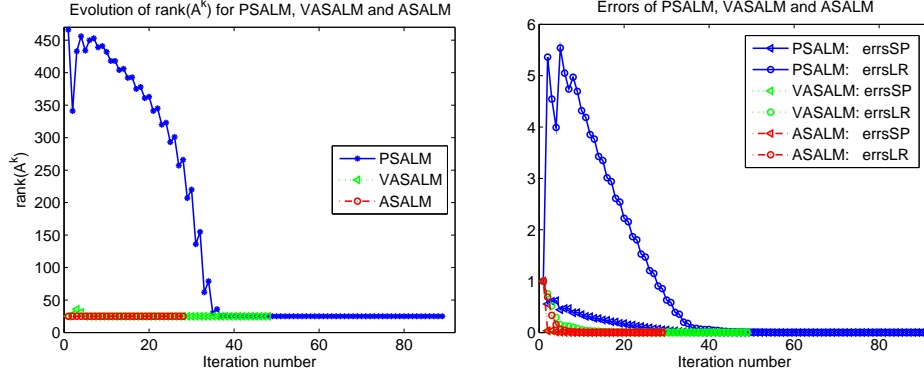


FIG. 9.1. Evolution of $\text{rank}(A^k)$ (left) and the errors of low rank and sparse components (right) for ASALM, VASALM and PSALM.

PSALM) destroys the underlying low-rank feature. On the opposite, the low-rank feature is well preserved by ASALM and VASALM; and this merit is very suitable for the application of PROPACK. In fact, this advantage shared by the ASALM and VASALM contributes much to the impressive numerical performance of them, as we shall report soon. For exposing the difference of these three 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 9.1. As shown, compared to those of ASALM and VASALM, the errors generated by PSALM's iterations again change more radically, especially at the beginning stage of iterations. Hence, in the implementation, we use PROPACK to execute partial SVD for ASALM and VASALM, while full SVD is executed for PSALM.

We also amplify parts of the curves of Figure 9.1 in Figure 9.2, in order to expose the different behaviors of these three splitting methods clearly.

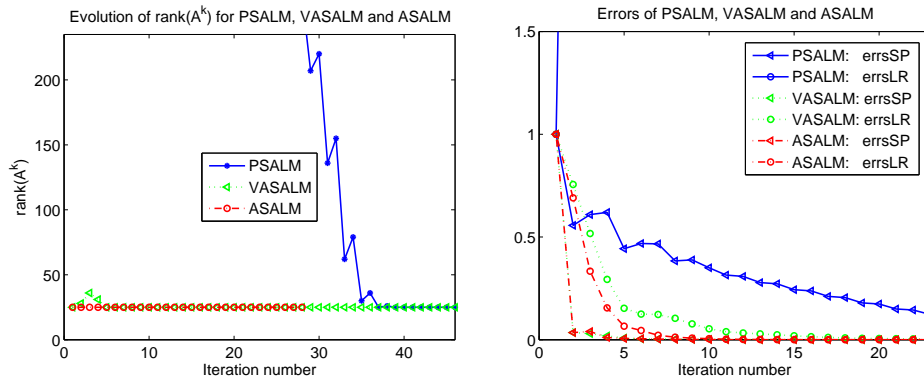


FIG. 9.2. Amplification of local zoom for Figure 9.1.

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 numeri-

TABLE 9.1
Numerical comparison of ASALM, VASALM and PSALM

$m = n = 500$														
rr	spr	sr	$\frac{\ \hat{E} - E^*\ _F}{\ E^*\ _F}$			$\frac{\ \hat{A} - A^*\ _F}{\ A^*\ _F}$			Time (s)			#SVD		
			PS	VA	AS	PS	VA	AS	PS	VA	AS	PS	VA	AS
0.05	0.05	0.9	2.24e-7	8.68e-6	2.87e-7	4.91e-6	4.29e-5	2.95e-6	80.1	15.4	11.5	89	38	26
		0.8	4.73e-7	5.18e-7	3.05e-7	1.16e-5	1.85e-5	3.67e-7	83.2	20.4	11.3	93	48	26
		0.7	3.17e-7	1.42e-5	2.47e-7	1.02e-5	9.14e-5	3.67e-6	96.9	23.7	13.0	110	59	33
		0.6	3.96e-7	8.76e-6	8.73e-6	1.05e-5	7.04e-5	6.29e-5	108.7	30.3	15.1	125	65	38
	0.1	0.9	4.25e-6	2.21e-5	6.29e-6	3.30e-5	1.51e-4	4.52e-5	132.5	14.8	15.0	148	35	25
		0.8	2.37e-6	2.22e-5	6.30e-6	2.44e-5	1.75e-4	5.27e-5	176.2	19.6	15.3	199	47	28
		0.7	3.96e-6	2.07e-5	6.16e-6	4.44e-5	1.96e-4	5.89e-5	180.8	23.5	17.4	206	50	35
		0.6	1.17e-5	2.58e-5	8.68e-6	2.92e-4	4.86e-4	1.03e-4	247.2	35.5	22.1	289	74	37
0.1	0.05	0.9	1.28e-6	8.50e-7	3.84e-7	1.76e-5	1.41e-5	3.05e-6	89.6	23.7	15.5	100	48	27
		0.8	1.06e-6	7.47e-7	3.27e-7	2.03e-5	1.82e-5	6.19e-6	91.5	32.6	19.4	103	60	32
		0.7	1.29e-6	9.75e-6	3.77e-7	2.25e-5	1.17e-4	7.79e-6	105.3	41.7	23.5	121	76	39
	0.1	0.9	5.28e-6	2.64e-5	9.47e-6	3.64e-5	1.85e-4	6.84e-5	132.6	24.6	17.7	149	49	28
		0.8	5.34e-6	9.60e-6	9.34e-6	4.46e-5	8.23e-5	7.81e-5	164.8	35.2	20.1	188	63	34
		0.7	1.63e-4	1.85e-4	1.73e-4	2.67e-3	3.04e-3	3.85e-3	738.2	137.4	112.0	841	164	135

TABLE 9.2
Parameter values of ASALM and VASALM

$m = n = 500$					
rr	spr	sr	VASALM		ASALM
			β	η	β
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	$0.25\beta_0$

cal performance. the numerical performance of ASALM (further abbreviated as AS), VASALM (VS) and PSALM (PS) are reported in Table 9.1. 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).

Note that for the data in Table 9.1, the involved parameters of these three methods are determined as follows. More specifically, let $\beta_0 = \frac{|\Omega|}{\|P_{\Omega}(C)\|_1}$, then the implementation of PSALM takes the fixed value of $\beta = 0.15 * \beta_0$ and $\gamma = 0.9$. For the implementation of ASALM and VASALM, for various scenarios, we take the value of β as suggested in Table 9.2. Note that the value of η for VASALM is also included in Table 9.2.

We must point out that both PSALM and VASALM should be more efficient than what we report in Table 9.2, as the subproblems that are eligible for parallel computation are actually solved in the consecutive order in our experiments. We

TABLE 9.3
Recovery results of ASALM for (1.4) with $\mathbf{sr} = 80\%$

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

believe that its efficiency can be improved if parallel computation facilities are available. Nevertheless, the preference of ASALM and VASALM, especially the ASALM, is shown clearly in Table 9.2. To witness the efficiency, we test more scenarios for ASALM and report the results in Table 9.3.

9.2. Gaussian-noisy case. Because of the obvious numerical efficiency of ASALM reported in the last subsection, and for the sake of succinctness, in this subsection we only apply the ASALM to solve the Gaussian-noisy case, i.e., (1.2). When the ASALM is implemented, the value of β is taken as $0.15 * \beta_0$ when $n > 200$ and $0.25 * \beta_0$ otherwise.

As clarified in [7], if the Gaussian noise is a white noise with standard deviation σ , then the parameter δ in (1.2) satisfies $\delta^2 \leq (d + \sqrt{8d})\sigma^2$ with high probability. Therefore, we set $\delta = \sqrt{d + \sqrt{8d}}\sigma$ in (1.2) when the ASALM is 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 (9.1) into

$$\text{RelChg} \triangleq \frac{\|(A^{k+1}, E^{k+1}) - (A^k, E^k)\|_F}{\|(A^k, E^k)\|_F + 1} \leq 0.1\sigma. \quad (9.2)$$

We generate the data C exactly as the last subsection, and report some numerical results of the ASALM for (1.2) for the scenarios that $m = n = 200, 500, 1000, 1500, 2000$, and $(rr, spr) = (5\%, 5\%), (5\%, 10\%), (10\%, 5\%)$, for the particular case where $\mathbf{sr} = 80\%$ and $\sigma = 10^{-3}$ in Table 9.4.

From Table 9.4, we see that the proposed model (1.2) is powerful for recovering low-rank and sparse components even for the Gaussian-noisy case; and that the proposed ASALM is very efficient for solving the model (1.2). In addition, to test the

TABLE 9.4
Recovery results of ASALM for (1.2) with $\text{sr} = 80\%$ and $\sigma = 10^{-3}$

n	rr	spr	$\frac{\ \hat{E}-E^*\ _F}{\ E^*\ _F}$	$\ \hat{E}\ _0$	$\frac{\ \hat{A}-A^*\ _F}{\ A^*\ _F}$	It.	Time (s)	#SVD
200	0.05	0.05	4.31e-5	1999	3.85e-4	15	1.1	15
		0.1	4.64e-5	4001	4.47e-4	16	1.2	17
	0.1	0.05	3.43e-5	2001	7.20e-4	16	2.0	19
		0.1	5.55e-4	7178	1.55e-2	67	6.2	67
500	0.05	0.05	3.62e-5	12499	6.52e-4	15	9.9	17
		0.1	4.64e-5	24984	5.97e-4	15	9.4	17
	0.1	0.05	4.40e-5	12501	7.07e-4	19	19.3	24
		0.1	7.18e-5	25179	9.28e-4	17	21.3	22
1000	0.05	0.05	3.05e-5	49982	4.32e-4	17	68.8	19
		0.1	6.31e-5	99915	6.46e-4	16	64.4	17
	0.1	0.05	4.17e-5	50195	4.03e-4	24	220.1	31
		0.1	5.58e-5	99996	7.16e-4	22	177.4	26
1500	0.05	0.05	1.87e-4	112595	2.75e-4	20	364.8	25
		0.1	4.04e-5	224898	5.22e-4	18	236.4	19
	0.1	0.05	4.40e-5	112503	4.13e-4	24	588.8	30
		0.1	6.48e-5	224975	6.88e-4	23	571.0	27
2000	0.05	0.05	2.32e-4	200104	2.64e-4	21	753.0	22
		0.1	4.55e-5	399915	5.37e-4	19	655.8	19
	0.1	0.05	4.24e-5	199992	3.42e-4	26	1398.3	29
		0.1	6.27e-5	399956	5.85e-4	25	1592.3	27

TABLE 9.5
Recovery results of ASALM for (1.2) with different δ 's

$m = n = 800$, and noise level $\sigma = 10^{-3}$, $\delta_0 = \sqrt{(d + \sqrt{8d})\sigma}$									
rr	spr	δ	RelChg	$\frac{\ \hat{E}-E^*\ _F}{\ E^*\ _F}$	$\ \hat{E}\ _0$	$\frac{\ \hat{A}-A^*\ _F}{\ A^*\ _F}$	It.	Time (s)	#SVD
0.05	0.05	$\delta_0/10$	8.97e-5	4.23e-5	31980	4.81e-4	16	34.3	17
		δ_0	8.97e-5	4.22e-5	31980	4.81e-4	16	34.5	17
		$10\delta_0$	8.96e-5	4.17e-5	31980	4.87e-4	16	34.7	17
0.05	0.1	$\delta_0/10$	7.82e-5	5.06e-5	63944	5.96e-4	16	40.1	18
		δ_0	7.82e-5	5.05e-5	63944	5.96e-4	16	40.4	18
		$10\delta_0$	7.81e-5	4.99e-5	63944	5.98e-4	16	39.6	18
0.1	0.05	$\delta_0/10$	8.83e-5	5.23e-5	31995	6.75e-4	21	76.6	25
		δ_0	8.83e-5	5.23e-5	31995	6.75e-4	21	75.5	25
		$10\delta_0$	8.81e-5	5.20e-5	31993	6.72e-4	21	76.6	25
0.1	0.01	$\delta_0/10$	8.54e-5	5.59e-5	64024	6.84e-4	20	81.9	23
		δ_0	8.54e-5	5.58e-5	64023	6.83e-4	20	81.4	23
		$10\delta_0$	8.53e-5	5.54e-5	64020	6.79e-4	20	81.2	23

sensitivity of the ASALM to the starting parameter δ , we test the ASALM for (1.2) with various δ in Table 9.5 for the particular scenario where $m = n = 800$, $\sigma = 10^{-3}$ and $\text{sr} = 80\%$.

Tables 9.4 and 9.5 show that the model (1.2) and the proposed ASALM are very robust and efficient, and they are easy to handle since a wide range of values of δ can lead to successful recovery.

9.3. The nuclear-norm- and l_1 -norm-regularized least squares case. In this section, we apply the extended ASALM for solving (8.1), i.e., the nuclear-norm-

TABLE 9.6
Recovery results of ASALM for (8.1) with $\text{sr} = 80\%$ and $\sigma = 10^{-3}$

n	rr	spr	$\frac{\ \hat{E}-E^*\ _F}{\ E^*\ _F}$	$\ \hat{E}\ _0$	$\frac{\ \hat{A}-A^*\ _F}{\ A^*\ _F}$	It.	Time (s)	#SVD
200	0.05	0.05	4.3e-5	1999	3.85e-4	15	1.2	15
		0.1	4.71e-5	4056	6.04e-4	17	1.3	17
	0.1	0.05	4.51e-5	2000	4.99e-4	16	2.1	19
		0.1	5.55e-4	7178	1.55e-2	67	6.1	67
500	0.05	0.05	3.62e-5	12499	6.52e-4	15	9.8	17
		0.1	4.64e-5	24984	5.97e-4	15	9.4	17
	0.1	0.05	4.40e-5	12501	7.07e-4	19	18.8	24
		0.1	7.18e-5	25179	9.28e-4	17	20.7	22
1000	0.05	0.05	3.05e-5	49982	4.32e-4	17	61.1	19
		0.1	6.31e-5	99915	6.45e-4	16	57.5	17
	0.1	0.05	4.17e-5	50190	4.03e-4	24	205.3	31
		0.1	5.58e-5	99996	7.15e-4	22	178.2	26
1500	0.05	0.05	1.87e-4	112591	2.75e-4	20	353.3	25
		0.1	4.04e-5	224898	5.22e-4	18	216.9	19
	0.1	0.05	4.40e-5	11250	4.13e-4	24	597.1	30
		0.1	6.47e-5	224973	6.88e-4	23	623.0	27
2000	0.05	0.05	2.32e-4	200098	2.64e-4	21	752.1	22
		0.1	4.56e-5	399915	5.38e-4	19	652.1	19
	0.1	0.05	4.25e-5	199989	3.43e-4	26	1355.0	29
		0.1	6.27e-5	399954	5.85e-4	25	1550.0	27

and l_1 -norm-regularized least square model for recovering low-rank and sparse components from incomplete and noisy observations.

We generate the data C exactly as what we have done in Section 9.1 and terminate the recovery once the criterion (9.2) is achieved. Again, when the extended ASALM is implemented, the value of β is taken as $0.15 * \beta_0$ when $n > 200$ and $0.25 * \beta_0$ otherwise. We set $\mu = \sqrt{d + \sqrt{8d\sigma}}/10$ in (8.1). In Table 9.6, we report some numerical results of the ASALM for (8.1) for the scenario that $m = n = 200, 500, 1000, 1500, 2000$, and $(\text{rr}, \text{spr}) = (5\%, 5\%), (5\%, 10\%), (10\%, 5\%)$, for the particular case where $\text{sr} = 80\%$ and $\sigma = 10^{-3}$.

As we can see in Table 9.6, low-rank and sparse components of matrices can also be recovered very well in many cases via solving (8.1) by the proposed ASALM.

Last, we test the ASALM for various starting parameter μ 's in (8.1) to witness the sensitivity to the value of μ . In particular, we choose $m = n = 800$, $\sigma = 10^{-3}$, $\text{sr} = 80\%$, and report the numerical results in Table 9.7.

Tables 9.6 and 9.7 together shows the attractive recovery ability of the model (8.1); and also the efficiency and robustness of the extended ASALM for solving this model.

10. Conclusions. This paper puts forward to 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 easily-implementable algorithms for solving the proposed model are developed based on the classical augmented Lagrangian method. 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 resulted recovery components are very accurate. Hence, the recovery model is justified empirically.

TABLE 9.7
Recovery results of ASALM for (8.1) with different μ 's

$m = n = 800$, and noise level $\sigma = 10^{-3}$, $\mu_0 = \sqrt{(d + \sqrt{8d})}\sigma/10$									
rr	spr	μ	RelChg	$\frac{\ \hat{E} - E^*\ _F}{\ E^*\ _F}$	$\ \hat{E}\ _0$	$\frac{\ \hat{A} - A^*\ _F}{\ A^*\ _F}$	It.	Time (s)	#SVD
0.05	0.05	$\mu_0/10$	8.96e-5	4.22e-5	31980	4.81e-4	16	30.2	17
		μ_0	8.95e-5	4.16e-5	31980	4.9e-4	16	30.1	17
		$10\mu_0$	8.81e-5	5.41e-5	31980	7.7e-4	16	29.4	17
0.05	0.1	$\mu_0/10$	7.82e-5	5.04e-5	63944	5.96e-4	16	42.3	18
		μ_0	7.81e-5	4.96e-5	63944	6.01e-4	16	42.0	18
		$10\mu_0$	7.72e-5	5.81e-5	63941	8.8e-4	16	41.7	18
0.1	0.05	$\mu_0/10$	8.83e-5	5.23e-5	31995	6.75e-4	21	79.4	25
		μ_0	8.79e-5	5.19e-5	31992	6.72e-4	21	79.3	25
		$10\mu_0$	8.44e-5	7.00e-5	31992	8.64e-4	21	78.6	25
0.1	0.01	$\mu_0/10$	8.54e-5	5.58e-5	64022	6.83e-4	20	85.2	23
		μ_0	8.52e-5	5.51e-5	64016	6.77e-4	20	84.8	23
		$10\mu_0$	8.36e-5	7.14e-5	63973	9.12e-4	20	83.7	23

As pointed out in [7], 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, *Aweak-to-strong convergence principle for Fejrmotone methods in Hilbert spaces*, Mathematics of Operations Research, 26(2) (2001), pp. 248-264.
- [2] D. P. BERTSEKAS, *Constrained Optimization and Lagrange Multiplier methods*, Academic Press, 1982.
- [3] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and distributed computation: Numerical methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [4] A. M. BRUCKSTEIN, D. L. DONOHO AND M. ELAD, *From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images*, SIAM Review, 51(1) (2009), pp. 34-81.
- [5] J. F. CAI, E. J. CANDÉS AND Z. W. SHEN, *A singular value thresholding algorithm for matrix completion*, preprint, available at <http://arxiv.org/abs/0810.3286>.
- [6] E. J. CANDÉS AND B. RECHT, *B. Recht, Exact Matrix Completion Via Convex Optimization*, Foundations of Computational Mathematics, 9 (2009), pp. 717-772.
- [7] E. J. CANDÉS AND Y. PLAN, *Matrix completion with noise*, preprint, available at http://arxiv.org/PS_cache/arxiv/pdf/0903/0903.3131v1.pdf.
- [8] E. J. CANDÉS AND T. TAO, *The power of convex relaxation: Near-Optimal matrix completion*, preprint, available at http://arxiv.org/PS_cache/arxiv/pdf/0906/0906.2027v1.pdf
- [9] V. CHANDRASEKARAN, S. SANGHAVI, P. A. PARRILO AND A. S. WILLSKYC, *Rank-sparsity incoherence for matrix decomposition*, manuscript, <http://arxiv.org/abs/0906.2220>.
- [10] G. CHEN, AND M. TEOULLE, *A proximal-based decomposition method for convex minimization problems*, Mathematical Programming, 64 (1994), pp. 81-101.
- [11] S. CHEN, D. DONOHO, AND M. SAUNDERS, *Atomic decomposition by basis pursuit*, SIAM Journal on Scientific Computing, 20 (1998), pp. 33-61.
- [12] J. DOUGLAS AND H. H. RACHFORD, *On the numerical solution of the heat conduction problem in 2 and 3 space variables*, Transactions of the American Mathematical Society, 82 (1956), pp. 421-439.

- [13] D. DONOHO, *High-dimensional data analysis: The curses and blessings of dimensionality*, AMS Math Challenges Lecture, 2000.
- [14] J. ECKSTEIN, *Some saddle-function splitting methods for convex programming*, Optimization Methods and Software, 4 (1994), pp. 75-83.
- [15] J. ECKSTEIN AND M. FUKUSHIMA, *Some reformulation and applications of the alternating directions method of multipliers*, In: Hager, W. W. et al. eds., Large Scale Optimization: State of the Art, Kluwer Academic Publishers, pp. 115-134, 1994.
- [16] E. ESSER, *Applications of Lagrangian-based alternating direction methods and connections to split Bregman*, Manuscript, <http://www.math.ucla.edu/applied/cam/>.
- [17] F. FACCHINEI AND J.-S. PANG, *Finite-Dimensional Variational Inequalities and Complementarity problems*, Volume I, Springer Series in Operations Research, Springer-Verlag, 2003.
- [18] M. FAZEL, *Matrix rank minimization with applications*, PhD thesis, Stanford University, 2002.
- [19] M. FAZEL AND J. GOODMAN, *Approximations for partially coherent optical imaging systems*, Technical Report, Stanford University, 1998.
- [20] M. FAZEL, H. HINDI, AND S. BOYD, *A rank minimization heuristic with application to minimum order system approximation*, Proceedings American Control Conference, 6 (2001), pp. 4734-4739.
- [21] M. FAZEL, H. HINDI AND S. BOYD, *Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices*, Proceedings of the American Control Conference, 2003.
- [22] M. FUKUSHIMA, *Application of the alternating direction method of multipliers to separable convex programming problems*, Computational Optimization and Applications, 1 (1992), pp. 93-111.
- [23] D. GABAY, *Application of the method of multipliers to variational inequalities*, In: Fortin, M., Glowinski, R., eds., Augmented Lagrangian methods: Application to the numerical solution of Boundary-Value Problem, North-Holland, Amsterdam, The Netherlands, pp. 299-331, 1983.
- [24] D. GABAY AND B. MERCIER, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Computer and Mathematics with Applications, 2 (1976), pp. 17-40.
- [25] R. GLOWINSKI, *Numerical methods for nonlinear variational problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.
- [26] R. GLOWINSKI AND P. LE TALLEC, *Augmented Lagrangian and Operator Splitting Methods in Nonlinear Mechanics*, SIAM Studies in Applied Mathematics, Philadelphia, PA, 1989.
- [27] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computation*, third edition, The Johns Hopkins University Press, 1996.
- [28] B. S. HE, *Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities*, Computational Optimization and Applications, 42 (2009), pp. 195-212.
- [29] B. S. HE, L. Z. LIAO, D. HAN AND H. YANG, *A new inexact alternating directions method for monotone variational inequalities*, Mathematical Programming, 92 (2002), pp. 103-118.
- [30] B. S. HE, M. H. XU AND X. M. YUAN, *Solving large-scale least squares covariance matrix problems by alternating direction methods*, manuscript, 2009.
- [31] B.S. HE AND H. YANG, *Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities*, Operations Research Letters, 23 (1998), pp. 151-161.
- [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, 1986.
- [34] R. H. KESHAVAN, A. MONTANARI AND S. OH, *Matrix completion from a few entries*, preprint, available at http://arxiv.org/PS_cache/arxiv/pdf/0901/0901.3150v4.pdf.
- [35] R. H. KESHAVAN, A. MONTANARI AND S. OH, *Matrix completion from noisy entries*, preprint, available at http://arxiv.org/PS_cache/arxiv/pdf/0906/0906.2027v1.pdf.
- [36] S. KONTOGIORGIS AND R. R. MEYER, *A variable-penalty alternating directions method for convex optimization*, Mathematical Programming, 83 (1998), pp. 29-53.
- [37] R. M. LARSEN, *Lanczos bidiagonalization with partial reorthogonalization*, Department of Computer Science, Aarhus University, Technical report, DAIMI PB-357, code available at <http://soi.stanford.edu/rmunk/PROPACK> (1998).

- [38] S. L. LAURITZEN, *Graphical Models*, Oxford University Press, 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, available at <http://perception.csl.illinois.edu/jnwright/Lin2009-Algorithms.pdf>.
- [41] S. Q. MA, D. GOLDFARB AND L. F. CHEN, *Fixed point and Bregman iterative methods for matrix rank minimization*, preprint, 2008.
- [42] M. NG, P. A. WEISS AND X. M. YUAN, *Solving constrained total-Variation problems via alternating direction methods*, Manuscript, 2009.
- [43] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*. Second Edition, Springer Verlag, 2006.
- [44] Y. C. PATI AND T. KAILATH, *Phase-shifting masks for microlithography: Automated design and mask requirements*, Journal of the Optical Society of America A, 11 (9), 1994.
- [45] B. RECHT, M. FAZEL AND P. A. PARRILO, *Guaranteed Minimum Rank Solutions to Linear Matrix Equations via Nuclear Norm Minimization*, SIAM Review, to appear.
- [46] E. SONTAG, *Mathematical Control Theory*, Springer-Verlag, New York, 1998.
- [47] J. F. STURM, *Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones*, Optimization Methods and Software 11 & 12 (1999), pp. 625-653.
- [48] J. SUN AND S. ZHANG, *A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs*, manuscript, 2009.
- [49] M. TAO AND J. F. YANG, *Alternating direction algorithms for total variation deconvolution in image reconstruction*, available at http://www.optimization-online.org/DB_HTML/2009/11/2463.html.
- [50] P. TSENG, *Alternating projection-proximal methods for convex programming and variational inequalities*, SIAM Journal on Optimization, 7 (1997), pp. 951-965.
- [51] R. H. TÜTÜNCÜ, K. C. TOH AND M. J. TODD, *Solving semidefinite-quadratic-linear programs using SDPT3*, Mathematical Programming, 95 (2003), pp. 189-217.
- [52] K. C. TOH AND S. YUN, *An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems*, preprint, available at <http://www.math.nus.edu.sg/matys/apg.pdf>.
- [53] L. G. VALIANT, *Graph-theoretic arguments in low-level complexity*, 6th Symposium on Mathematical Foundations of Computer Science, (1977), pp. 162-176.
- [54] J. WRIGHT, A. GANESH, S. RAO AND Y. MA, *Robust Principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization*, manuscript, 2009.
- [55] Z. W. WEN, D. GOLDFARB AND W. YIN, *Alternating direction augmented lagrangina methods for semidefinite programming*, manuscript, 2009.
- [56] J. F. YANG, W. T. YIN, Y. ZHANG, AND Y. L. WANG, *A fast algorithm for edge-preserving variational multichannel image Rstoration*, SIAM Journal on Imaging Science, 2 (2009), pp. 569-592.
- [57] J. YANG AND Y. ZHANG, *Alternating direction algorithms for l1-problems in compressive sensing*, TR09-37, CAAM, Rice University.
- [58] C. H. YE AND X. M. YUAN, *A Descent Method for structured monotone variational inequalities*, Optimization Methods and Software, 22 (2007), pp. 329-338.
- [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.