

# A Unified Alternating Direction Method of Multipliers by Majorization Minimization

Canyi Lu, *Student Member, IEEE*, Jiashi Feng, Shuicheng Yan, *Senior Member, IEEE*, and Zhouchen Lin, *Senior Member, IEEE*

**Abstract**—Accompanied with the rising popularity of compressed sensing, the Alternating Direction Method of Multipliers (ADMM) has become the most widely used solver for linearly constrained convex problems with separable objectives. In this work, we observe that many previous variants of ADMM update the primal variable by minimizing different majorant functions with their convergence proofs given case by case. Inspired by the principle of majorization minimization, we respectively present the unified frameworks and convergence analysis for the Gauss-Seidel ADMMs and Jacobian ADMMs, which use different historical information for the current updating. Our frameworks further generalize previous ADMMs to the ones capable of solving the problems with non-separable objectives by minimizing their separable majorant surrogates. We also show that the bound which measures the convergence speed of ADMMs depends on the tightness of the used majorant function. Then several techniques are introduced to improve the efficiency of ADMMs by tightening the majorant functions. In particular, we propose the Mixed Gauss-Seidel and Jacobian ADMM (M-ADMM) which alleviates the slow convergence issue of Jacobian ADMMs by absorbing merits of the Gauss-Seidel ADMMs. M-ADMM can be further improved by using backtracking, wise variable partition and fully exploiting the structure of the constraint. Beyond the guarantee in theory, numerical experiments on both synthesized and real-world data further demonstrate the superiority of our new ADMMs in practice. Finally, we release a toolbox at <https://github.com/canyilu/LibADMM> that implements efficient ADMMs for many problems in compressed sensing.

**Index Terms**—Alternating Direction Method of Multipliers, Majorization Minimization, Convex Optimization

## 1 INTRODUCTION

THIS work aims to solve the following convex problem

$$\min_{\mathbf{x}} f(\mathbf{x}) = f(\mathbf{x}_1, \dots, \mathbf{x}_n), \text{ s.t. } \mathbf{Ax} = \sum_{i=1}^n \mathbf{A}_i \mathbf{x}_i = \mathbf{b}, \quad (1)$$

where  $f : \mathbb{R}^{p_1 \times \dots \times p_n} \rightarrow \mathbb{R}$  is convex and  $n (\geq 2)$  denotes the block number of variables. We denote  $\mathbf{x} = [\mathbf{x}_1; \dots; \mathbf{x}_n]$  with  $\mathbf{x}_i \in \mathbb{R}^{p_i}$ , and  $\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_n]$  with  $\mathbf{A}_i \in \mathbb{R}^{d \times p_i}$ . Problem (1) has drawn increasing attention recently for the emerging applications of compressive sensing in computer vision and signal processing, e.g., sparsity based face recognition [40], [7], saliency detection [36], motion segmentation [25], [28], [8], image denoising [11], [22], video denoising [17], texture repairing [19] and many others [5], [42], [38], [41], [16].

To solve (1), the popular Augmented Lagrangian Method (ALM) [14] updates the primal variable  $\mathbf{x}$  by

$$\mathbf{x}^{k+1} = \arg \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^k, \beta^{(k)}) = \arg \min_{\mathbf{x}} f(\mathbf{x}) + r^k(\mathbf{x}), \quad (2)$$

where  $\mathcal{L}$  is the augmented Lagrangian function defined as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \beta) = f(\mathbf{x}) + \langle \boldsymbol{\lambda}, \mathbf{Ax} - \mathbf{b} \rangle + \frac{\beta}{2} \|\mathbf{Ax} - \mathbf{b}\|^2,$$

and

$$r^k(\mathbf{x}) = \frac{\beta^{(k)}}{2} \left\| \mathbf{Ax} - \mathbf{b} + \frac{\boldsymbol{\lambda}^k}{\beta^{(k)}} \right\|^2. \quad (3)$$

Then the dual variable  $\boldsymbol{\lambda}$  is updated to minimize  $-\mathcal{L}$  by gradient descent with the step size  $\beta^{(k)}$ , i.e.,

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta^{(k)} (\mathbf{Ax}^{k+1} - \mathbf{b}). \quad (4)$$

However, (2) may not be easily solvable, since  $r^k$  is non-separable. The Alternating Direction Method of Multipliers (ADMM) [9] instead solves (2) inexactly by updating  $\mathbf{x}_i$ 's in an alternating way and thus the per-iteration cost can be much lower. Many variants of ADMM have been proposed by using different properties of  $f$  and  $\mathbf{A}$ . We will review the most related works in Section 1.1, and claim our contributions in Section 1.2.

**Notations.** The  $\ell_2$ -norm of a vector and Frobenius norm of a matrix are denoted as  $\|\cdot\|$ . The spectral norm and the smallest singular value of a matrix  $\mathbf{A}$  are denoted as  $\|\mathbf{A}\|_2$  and  $\sigma_{\min}(\mathbf{A})$ , respectively. The identity matrix is denoted as  $\mathbf{I}$  without specifying its size. The all-one vector is denoted as  $\mathbf{1}$ . We denote  $\mathbb{S}$  and  $\mathbb{S}_+$  as the set of symmetry and positive semidefinite matrices respectively and define  $\langle \mathbf{a}, \mathbf{a} \rangle_{\mathbf{A}} = \|\mathbf{a}\|_{\mathbf{A}}^2 = \mathbf{a}^\top \mathbf{A} \mathbf{a}$  for  $\mathbf{A} \in \mathbb{S}$ . If  $\mathbf{A} - \mathbf{B}$  is positive semi-definite, then we denote  $\mathbf{A} \succeq \mathbf{B}$ . The block diagonal matrix  $\text{Diag}\{\mathbf{A}_i, i = 1, \dots, n\}$  has  $\mathbf{A}_i$  as its  $i$ -th block on the diagonal. A function  $f : \mathbb{R}^p \rightarrow \mathbb{R}$  is said to be  $L$ -smooth (or  $\nabla f$  is Lipschitz continuous), if

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p. \quad (5)$$

### 1.1 Review of ADMMs

Most of ADMMs are only able to solve (1) with separable  $f$ ; i.e., there exist  $f_i$ 's such that  $f(\mathbf{x}) = \sum_{i=1}^n f_i(\mathbf{x}_i)$ . They can be categorized into Gauss-Seidel ADMMs and Jacobian ADMMs. The Gauss-Seidel ADMMs update  $\mathbf{x}_i$ 's in a sequential way, i.e., update  $\mathbf{x}_i^{k+1}$  by fixing others as their latest versions, while the Jacobian ADMMs update  $\mathbf{x}_i$ 's in a parallel way, i.e., update each

- C. Lu, J. Feng and S. Yan are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore (e-mail: canyilu@gmail.com; elefjia@nus.edu.sg; eleyans@nus.edu.sg).
- Z. Lin is with the Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, China (e-mail: zlin@pku.edu.cn).

$\mathbf{x}_i^{k+1}$  by fixing  $\mathbf{x}_j = \mathbf{x}_j^k$ , for all  $j \neq i$ . We review these two types of ADMMs respectively. The difference between ADMMs lies in the updating of  $\mathbf{x}_i$ 's, while  $\boldsymbol{\lambda}$  is updated in the same way by (4).

Gauss-Seidel ADMMs solve (1) with  $n = 2$  blocks. The standard ADMM [2] solves (2) inexactly by updating  $\mathbf{x}_1$  and  $\mathbf{x}_2$  in a sequential way, i.e.,

$$\begin{aligned} \mathbf{x}_1^{k+1} &= \arg \min_{\mathbf{x}_1} \mathcal{L}([\mathbf{x}_1; \mathbf{x}_2^k], \boldsymbol{\lambda}^k, \beta^{(k)}) \\ &= \arg \min_{\mathbf{x}_1} f_1(\mathbf{x}_1) + r_1^k(\mathbf{x}_1), \end{aligned} \quad (6)$$

$$\begin{aligned} \mathbf{x}_2^{k+1} &= \arg \min_{\mathbf{x}_2} \mathcal{L}([\mathbf{x}_1^{k+1}; \mathbf{x}_2], \boldsymbol{\lambda}^k, \beta^{(k)}) \\ &= \arg \min_{\mathbf{x}_2} f_2(\mathbf{x}_2) + r_2^k(\mathbf{x}_2), \end{aligned} \quad (7)$$

where

$$r_1^k(\mathbf{x}_1) = \frac{\beta^{(k)}}{2} \left\| \mathbf{A}_1 \mathbf{x}_1 + \mathbf{A}_2 \mathbf{x}_2^k - \mathbf{b} + \frac{\boldsymbol{\lambda}^k}{\beta^{(k)}} \right\|^2, \quad (8)$$

$$r_2^k(\mathbf{x}_2) = \frac{\beta^{(k)}}{2} \left\| \mathbf{A}_1 \mathbf{x}_1^{k+1} + \mathbf{A}_2 \mathbf{x}_2 - \mathbf{b} + \frac{\boldsymbol{\lambda}^k}{\beta^{(k)}} \right\|^2. \quad (9)$$

By using different properties of  $f_1$  and  $\mathbf{A}_1$ ,  $\mathbf{x}_1$  (the same discussion is also applicable to  $\mathbf{x}_2$ ) can be updated more efficiently than solving (6). If  $f_1$  is  $L_1$ -smooth, then  $\mathbf{x}_1$  can be updated by

$$\mathbf{x}_1^{k+1} = \arg \min_{\mathbf{x}_1} \hat{f}_1(\mathbf{x}_1) + r_1^k(\mathbf{x}_1), \quad (10)$$

where  $\hat{f}_1(\mathbf{x}_1) = f(\mathbf{x}_1^k) + \langle \nabla f_1(\mathbf{x}_1^k), \mathbf{x}_1 - \mathbf{x}_1^k \rangle + \frac{L_1}{2} \|\mathbf{x}_1 - \mathbf{x}_1^k\|^2$ . The motivation is that  $\hat{f}_1$  is a majorant (upper bound) function of  $f_1$ , i.e.,  $\hat{f}_1 \geq f_1$  [1]. If  $f_1 = g_1 + h_1$ , where  $g_1$  is convex and  $h_1$  is convex and  $L_1$ -smooth, then  $\mathbf{x}_1$  can be updated by (10) with  $\hat{f}_1(\mathbf{x}_1) = g(\mathbf{x}_1) + h(\mathbf{x}_1^k) + \langle \nabla h_1(\mathbf{x}_1^k), \mathbf{x}_1 - \mathbf{x}_1^k \rangle + \frac{L_1}{2} \|\mathbf{x}_1 - \mathbf{x}_1^k\|^2$ . In this case,  $\hat{f}_1 \geq f_1$ . We name the method using (10) as Proximal ADMM (P-ADMM) for these two cases. Similar techniques have been used in [1], [34].

If the columns of  $\mathbf{A}_1$  are not orthogonal, solving (6) is usually very expensive especially when  $f_1$  is nonsmooth. Then Linearized ADMM (L-ADMM) [23] instead updates  $\mathbf{x}_1$  by

$$\mathbf{x}_1^{k+1} = \arg \min_{\mathbf{x}_1} f_1(\mathbf{x}_1) + \hat{r}_1^k(\mathbf{x}_1), \quad (11)$$

where  $\hat{r}_1^k(\mathbf{x}_1) = r_1^k(\mathbf{x}_1^k) + \langle \nabla r_1^k(\mathbf{x}_1^k), \mathbf{x}_1 - \mathbf{x}_1^k \rangle + \frac{\eta_1}{2} \|\mathbf{x}_1 - \mathbf{x}_1^k\|^2$  with  $\eta_1 > \|\mathbf{A}_1\|_2^2$ . Note that  $\hat{r}_1^k \geq r_1^k$  since  $r_1^k$  is  $\|\mathbf{A}_1\|_2^2$ -smooth. For some nonsmooth  $f_1$ , e.g., the  $\ell_1$ -norm, (11) can be solved efficiently with a closed form solution.

If  $f_1$  is a sum of a nonsmooth function and an  $L_1$ -smooth function, then we can simultaneously use the majorant function  $\hat{f}_1$  of  $f_1$  as P-ADMM and  $\hat{r}_1^k$  of  $r_1^k$  as L-ADMM. Thus  $\hat{f}_1 + \hat{r}_1^k \geq f_1 + r_1^k$ . This motivates the Proximal Linearized ADMM (PL-ADMM) which updates  $\mathbf{x}_1$  by

$$\mathbf{x}_1^{k+1} = \arg \min_{\mathbf{x}_1} \hat{f}_1(\mathbf{x}_1) + \hat{r}_1^k(\mathbf{x}_1). \quad (12)$$

For (1) with  $n > 2$  blocks of variables, the naive extension of Gauss-Seidel ADMMs may diverge [3]. To address this issue, several Jacobian ADMMs have been proposed by using different properties of  $f_i$  and  $\mathbf{A}_i$ . The Linearized ADMM with Parallel Splitting (L-ADMM-PS) [27] solves (2) inexactly by linearizing  $r^k$  in (3) at  $\mathbf{x}_i^k$ 's and updates  $\mathbf{x}_i$ 's in parallel by

$$\begin{aligned} \mathbf{x}_i^{k+1} &= \arg \min_{\mathbf{x}_i} f_i(\mathbf{x}_i) + \left\langle \mathbf{A}_i^\top (\beta^{(k)} (\mathbf{A} \mathbf{x}^k - \mathbf{b}) + \boldsymbol{\lambda}^k), \mathbf{x}_i \right\rangle \\ &\quad + \frac{\beta^{(k)} \eta_i}{2} \|\mathbf{x}_i - \mathbf{x}_i^k\|^2, \end{aligned} \quad (13)$$

where  $\eta_i > n \|\mathbf{A}_i\|_2^2$ . A more general method proposed in the Algorithm 4 of [6] updates  $\mathbf{x}_i$ 's in parallel by

$$\begin{aligned} \mathbf{x}_i^{k+1} &= \arg \min_{\mathbf{x}_i} f_i(\mathbf{x}_i) + \frac{\beta^{(k)}}{2} \left\| \mathbf{A}_i \mathbf{x}_i + \sum_{j \neq i} \mathbf{A}_j \mathbf{x}_j^k - \mathbf{b} - \frac{\boldsymbol{\lambda}^k}{\beta^{(k)}} \right\|^2 \\ &\quad + \frac{\beta^{(k)}}{2} \|\mathbf{x}_i - \mathbf{x}_i^k\|_{\mathbf{G}_i}^2, \end{aligned} \quad (14)$$

where  $\mathbf{G}_i \succ (n-1) \mathbf{A}_i^\top \mathbf{A}_i$ . Actually (13) is a special case of (14) when  $\mathbf{G}_i = \eta_i \mathbf{I} - \mathbf{A}_i^\top \mathbf{A}_i$  with  $\eta_i > n \|\mathbf{A}_i\|_2^2$ . So we name the method using (14) as Generalized Linearized ADMM with Parallel Splitting (GL-ADMM-PS) in this work. If  $f_i = g_i + h_i$ , where  $g_i$  is convex and  $h_i$  is convex and  $L_i$ -smooth, then the Proximal Linearized ADMM with Parallel Splitting (PL-ADMM-PS) [22] updates  $\mathbf{x}_i$ 's in parallel by

$$\begin{aligned} \mathbf{x}_i^{k+1} &= \arg \min_{\mathbf{x}_i} \hat{f}_i(\mathbf{x}_i) + \left\langle \mathbf{A}_i^\top (\beta^{(k)} (\mathbf{A} \mathbf{x}^k - \mathbf{b}) + \boldsymbol{\lambda}^k), \mathbf{x}_i \right\rangle \\ &\quad + \frac{\beta^{(k)} \eta_i}{2} \|\mathbf{x}_i - \mathbf{x}_i^k\|^2, \end{aligned} \quad (15)$$

where  $\hat{f}_i(\mathbf{x}_i) = g(\mathbf{x}_i) + h(\mathbf{x}_i^k) + \langle \nabla h_i(\mathbf{x}_i^k), \mathbf{x}_i - \mathbf{x}_i^k \rangle + \frac{L_i}{2} \|\mathbf{x}_i - \mathbf{x}_i^k\|^2$  and  $\eta_i > n \|\mathbf{A}_i\|_2^2$ . As we will show later, the updating rules (14) and (15) are equivalent to minimizing different majorant functions of  $f(\mathbf{x}) + r^k(\mathbf{x})$  in (2).

For the convergence guarantee, all the above ADMMs own the convergence rate  $O(1/K)$  [12], [27], [22], where  $K$  is the number of iterations. There are also some other works which consider different special cases of our problem (1) and give different convergence rates of ADMMs. For example, the works [10], [29] propose fast ADMMs with better convergence rate. But their considered problems are quite specific and their convergence guarantees require several additional assumptions. For problem (1) with separable objective and  $n > 2$ , the works [15], [20], [21] prove the convergence of the naive multi-blocks extension of ADMM under various assumptions, e.g., full column rank of  $\mathbf{A}_i$ , strong convexity or Lipschitz continuity of some  $f_i$  and some others which may be hard to be verified in practice. The work [39] reformulates the multi-blocks problem into a two-block one by variable splitting and solves it by ADMM. But it is verified to be slower than GL-ADMM-PS in [6] since the variable splitting substantially increases the number of variables and constraints, especially when  $n$  is large.

## 1.2 Contributions

From the above discussions, we observe that different ADMMs can be regarded as variants of inexact ALM in the sense that the primal variable  $\mathbf{x}^{k+1}$  in ADMMs is updated by solving (2) in ALM approximately. This actually slows the convergence, but the per-iteration cost is lower. So there is a trade-off between the exactness of the subproblem optimization and the convergence speed. In practice, we balance both to choose the proper solver. Generally, if  $f$  is not very simple, e.g., sum of several nonsmooth functions, ADMMs are much more efficient than ALM. ADMMs use two main techniques for approximation and update  $\mathbf{x}^{k+1}$  in an easier way than ALM: Alternating Minimization (AM) and Majorization Minimization (MM) [18]. AM, which updates one block each time when fixing others, makes the subproblems easier to solve. For example, the updating of  $[\mathbf{x}_1^{k+1}; \mathbf{x}_2^{k+1}]$  in ADMM (6)-(7) is easier than the one in ALM (2). But the cost of the one block updating may be still high and it can be further reduced by

using MM, which minimizes a majorant function instead of the original objective to find an approximated solution. For example, as reviewed in Section 1.1, different Gauss-Seidel ADMMs update  $\mathbf{x}_1$  by minimizing different majorant functions of the objective in standard ADMM (6), while different Jacobian ADMMs update  $\mathbf{x}_i$ 's by minimizing different majorant functions of the objective in ALM (2). Actually, Gauss-Seidel ADMMs first use AM and then apply MM to update each block, while Jacobian ADMMs first use MM and then AM to update each block (though this is equivalent to updating all blocks simultaneously). Besides the primal variables, the dual variable  $\boldsymbol{\lambda}^{k+1}$  updating in (4) is also equivalent to minimizing a majorant function of  $-\mathcal{L}(\mathbf{x}^{k+1}, \boldsymbol{\lambda}, \beta^{(k)})$ , i.e.,

$$\boldsymbol{\lambda}^{k+1} = \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} -\mathcal{L}(\mathbf{x}^{k+1}, \boldsymbol{\lambda}, \beta^{(k)}) + \frac{1}{2\beta^{(k)}} \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^k\|^2. \quad (16)$$

These observations suggest that MM provides a new insight to interpret ADMMs. The convergences of ADMMs which use different majorant functions are guaranteed, but they are proved case by case. It is not clear what is the role of MM in ADMMs. Another issue is that, in practice, one can develop many ADMMs for the same problem. But it is generally difficult to see which one converges faster. The proved same rate  $O(1/K)$  in the worst case fails to characterize the different speeds of ADMMs in practice. We lack practical principles and guidelines for designing efficient ADMMs.

In this work, we raise several crucial questions:

- What kind of majorant functions can be used in ADMMs?
- Is that possible to give a unified convergence analysis of ADMMs which use different majorant functions by using certain common properties of majorant functions?
- What is the connection between the convergence speed of ADMMs and the used majorant functions?
- How to choose the proper majorant functions for designing efficient ADMMs?

In this work, we show many interesting findings about ADMMs through the lens of MM. We aim to address the above questions and in particular we make the following contributions. First, for a multivariable function  $f$ , we propose the majorant first-order surrogate function  $\hat{f}$ , which requires three conditions to be satisfied: majorization, proximity and separability. The first two guarantee that  $\hat{f}$  is a reasonable approximation of  $f$ , while the last one makes the minimizing of  $\hat{f}$  easy. Note that the objective  $f$  in (1) can be non-separable since we only need to minimize  $\hat{f}$ . Second, we present the unified frameworks of Gauss-Seidel ADMMs and Jacobian ADMMs based on our majorant first-order surrogate and give the unified convergence guarantee. They not only draw connections with existing ADMMs, but also extend them to solve new problems with non-separable objective. Third, we show that the bound which measures the convergence speed of ADMMs depends on the tightness of the used majorant function. The tighter, the faster. This explains our previous intuitive observation that ADMMs converge faster when (2) in ALM is solved more accurately. Fourth, we develop several useful techniques to tighten the majorant surrogates and thus improve the efficiency of ADMMs. Consider (1) with  $n > 2$ , we propose the Mixed Gauss-Seidel and Jacobian ADMM (M-ADMM) algorithm. It divides  $n$  blocks of variables into two super blocks, and then updates them in a sequential way as Gauss-Seidel ADMMs, while the variables in each super block are updated in a parallel way as Jacobian ADMMs. M-ADMM takes the structure

of  $\mathbf{A}$ , e.g.,  $\frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$  that may be partially separable, into account to compute a tighter majorant surrogate, while previous Jacobian ADMMs fail to do so. In addition, we show how to partition  $n$  blocks of variables into two super blocks wisely, which is crucial in the efficient implementation of ADMMs. The last contribution is the developed toolbox which implements efficient ADMMs for many popular problems in compressed sensing. See <https://github.com/canyilu/LibADMM>.

Though there are already many toolboxes in compressed sensing, the solved problems are more or less limited due to the applicability of the used solvers, e.g., SPAMS [32] and SLEP [26] focus more on sparse models and non-constrained problems. We instead focus on the constrained problem (1), which is much more general. See a list of problems in our toolbox in the supplementary material.

## 2 MAJORANT FIRST-ORDER SURROGATE OF A MULTIVARIABLE FUNCTION

In this section, we propose the majorant first-order surrogate of the multivariable functions which enjoy some “good” properties.

**Definition 1. (Lipschitz Continuity)** Let  $f : \mathbb{R}^{p_1} \times \cdots \times \mathbb{R}^{p_n} \rightarrow \mathbb{R}$  be differentiable. Then  $\nabla f$  is called Lipschitz continuous if there exist  $\mathbf{L}_i \succeq \mathbf{0}$ ,  $i = 1, \dots, n$ , such that

$$|f(\mathbf{x}) - f(\mathbf{y}) - \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle| \leq \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{L}_i}^2, \quad (17)$$

for any  $\mathbf{x} = [\mathbf{x}_1; \cdots; \mathbf{x}_n]$  and  $\mathbf{y} = [\mathbf{y}_1; \cdots; \mathbf{y}_n]$  with  $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^{p_i}$ . In this case, we say that  $f$  is  $\{\mathbf{L}_i\}_{i=1}^n$ -smooth.

The Lipschitz continuity of the multivariable function is crucial in this work. It is different from the single variable case defined in (5). For  $n = 1$ , (17) holds if (5) holds (Lemma 1.2.3 in [33]), but not vice versa. This motivates the above definition.

**Definition 2. (Strong Convexity)** A function  $f : \mathbb{R}^{p_1} \times \cdots \times \mathbb{R}^{p_n} \rightarrow \mathbb{R}$  is called  $\{\mathbf{P}_i\}_{i=1}^n$ -strongly convex if there exist  $\mathbf{P}_i \succeq \mathbf{0}$ ,  $i = 1, \dots, n$ , such that for any  $\mathbf{y}_i \in \mathbb{R}^{p_i}$ , the function  $\mathbf{x} \rightarrow f(\mathbf{x}) - \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{P}_i}^2$  is convex.

**Definition 3. (Majorant First-Order Surrogate)** A function  $\hat{f} : \mathbb{R}^{p_1} \times \cdots \times \mathbb{R}^{p_n} \rightarrow \mathbb{R}$  is a majorant first-order surrogate of  $f : \mathbb{R}^{p_1} \times \cdots \times \mathbb{R}^{p_n} \rightarrow \mathbb{R}$  near  $\boldsymbol{\kappa} = [\boldsymbol{\kappa}_1; \cdots; \boldsymbol{\kappa}_n]$  with  $\boldsymbol{\kappa}_i \in \mathbb{R}^{p_i}$  when the following conditions are satisfied:

- **Majorization:**  $\hat{f}$  is a majorant function of  $f$ , i.e.,  $\hat{f}(\mathbf{x}) \geq f(\mathbf{x})$  for any  $\mathbf{x}$ .
- **Proximity:** there exists  $\mathbf{L}_i \succeq \mathbf{0}$  such that the approximation error  $h(\mathbf{x}) := \hat{f}(\mathbf{x}) - f(\mathbf{x})$  satisfies

$$|h(\mathbf{x})| \leq \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \boldsymbol{\kappa}_i\|_{\mathbf{L}_i}^2. \quad (18)$$

- **Separability:**  $\hat{f}$  is separable w.r.t.  $\mathbf{x}_i$ 's; i.e., there exist  $\hat{f}_i$ 's such that  $\hat{f}(\mathbf{x}) = \sum_{i=1}^n \hat{f}_i(\mathbf{x}_i)$ .

We denote by  $\mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \boldsymbol{\kappa})$  the set of  $\{\mathbf{P}_i\}_{i=1}^n$ -strongly convex surrogates.

In MM, one aim to find an approximated solution to  $\min_{\mathbf{x}} f(\mathbf{x})$  by solving  $\min_{\mathbf{x}} \hat{f}(\mathbf{x})$ , which is easier. To this end, the above three conditions on  $\hat{f}$  look reasonable. Majorization guarantees that  $f(\mathbf{x})$  tends to be minimized when  $\hat{f}(\mathbf{x})$  is minimized. Proximity means that  $\hat{f}(\mathbf{x})$  cannot be too loose and this

guarantees a controllable approximation to  $f(\mathbf{x})$ . The separability makes the optimization on  $\hat{f}(\mathbf{x})$  easier than  $f(\mathbf{x})$ , which can be non-separable. This is important for multi-blocks optimization.

Note that  $\mathbf{L}_i$  measures the difference  $\hat{f} - f$ , or the tightness of the majorant surrogate  $\hat{f}$ . If  $\|\mathbf{L}_i\|_2$  is smaller, then the majorant surrogate is tighter. This plays an important role in this work.

**Lemma 1.** *If the approximation error  $h(\mathbf{x}) = \hat{f}(\mathbf{x}) - f(\mathbf{x})$  satisfies the following **Smoothness** assumption, i.e.,*

$$h(\mathbf{x}) \text{ is } \{\mathbf{L}_i\}_{i=1}^n\text{-smooth, } h(\boldsymbol{\kappa}) = 0 \text{ and } \nabla h(\boldsymbol{\kappa}) = 0, \quad (19)$$

then the **Proximity** assumption in (18) holds.

Lemma 1 can be obtained by using (17) for  $h$  at  $\boldsymbol{\kappa}$ . Lemma 1 is useful to verify the Proximity assumption. Some widely used majorant first-order surrogates are (see Lemma 5 in Appendix):

- **Proximal Surrogates.** For any  $f$  and  $\mathbf{L} \succeq \mathbf{0}$ ,  $\hat{f} \in \mathcal{S}_{\{\mathbf{L}, \mathbf{L}\}}(f, \boldsymbol{\kappa})$ , where  $\hat{f}(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \boldsymbol{\kappa}\|_{\mathbf{L}}^2$ .
- **Lipschitz Gradient Surrogates.** Let  $f$  be  $\{\mathbf{L}_i\}_{i=1}^n$ -smooth. Then  $\hat{f} \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{L}_i\}_{i=1}^n}(f, \boldsymbol{\kappa})$ , where  $\hat{f}(\mathbf{x}) = f(\boldsymbol{\kappa}) + \langle \nabla f(\boldsymbol{\kappa}), \mathbf{x} - \boldsymbol{\kappa} \rangle + \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \boldsymbol{\kappa}_i\|_{\mathbf{L}_i}^2$ .
- **Proximal Gradient Surrogates.** Let  $f = f_1 + f_2$ , where  $f_1$  is  $\{\mathbf{L}_i\}_{i=1}^n$ -smooth. Then  $\hat{f} \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{L}_i\}_{i=1}^n}(f, \boldsymbol{\kappa})$ , where  $\hat{f}(\mathbf{x}) = f_1(\boldsymbol{\kappa}) + \langle \nabla f_1(\boldsymbol{\kappa}), \mathbf{x} - \boldsymbol{\kappa} \rangle + \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \boldsymbol{\kappa}_i\|_{\mathbf{L}_i}^2 + f_2(\mathbf{x})$ .

Note that if  $f$  is separable, then  $\hat{f} = f$  is also a majorant first-order surrogate of  $f$ . Some other examples, e.g., DC programming surrogates, can be found in [31].

**Lemma 2. (Key Property of the Majorant First-Order Surrogate)** *Let  $\hat{f} \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \boldsymbol{\kappa})$ . Then, we have*

$$f(\mathbf{x}) + \langle \mathbf{u}, \mathbf{y} - \mathbf{x} \rangle - f(\mathbf{y}) \leq \frac{1}{2} \sum_{i=1}^n (\|\mathbf{y}_i - \boldsymbol{\kappa}_i\|_{\mathbf{L}_i}^2 - \|\mathbf{y}_i - \mathbf{x}_i\|_{\mathbf{P}_i}^2), \quad \forall \mathbf{x}, \mathbf{y}, \quad (20)$$

where  $\mathbf{u} \in \partial \hat{f}(\mathbf{x})$  is any subgradient of the convex  $\hat{f}$ .

The majorant first-order surrogate given in Definition 3 is motivated by [31]. However, they have many key differences:

- Our majorant first-order surrogate is defined based on the multivariable function and thus it is much more general than the single variable case considered in [31]. For example, the Lipschitz continuity of the multivariable function is different; the **Separability** of  $\hat{f}$  is new.
- For approximation error  $h = \hat{f} - f$ , we use the **Proximity** assumption in (18) which is less restricted than of the **Smoothness** assumption in (19). We only require the error  $h$  to be bounded, and it is not necessary to be smooth.
- Our Lemma 2 is new and it plays a central role in our convergence analysis. Lemma 2.1 in [31] also introduces some properties of the majorant first-order surrogate. But their bounds are too loose and are not applicable to our proofs due to the constraint of (1) considered in this work.
- The considered constrained problem in this work is different from the non-constrained problem in [31]. When proving Proposition 2.3 in [31], they use a key property  $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k)$ , while this does not hold in ADMMs.

At the end of this section, we discuss some properties of  $\frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|^2$  which are important for designing efficient ADMMs.

**Lemma 3.** *Let  $r(\mathbf{x}) = \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|^2$ , where  $\mathbf{x} = [\mathbf{x}_1; \dots; \mathbf{x}_n]$ ,  $\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_n]$  and  $\mathbf{b}$  are of compatible sizes. We have*

- (1)  $r(\mathbf{x})$  is  $\{\mathbf{L}'_i\}_{i=1}^n$ -smooth. The choice of  $\mathbf{L}'_i$  depends on  $\mathbf{A}_i^\top \mathbf{A}_i$ .
- (2)  $r(\mathbf{x}) \leq \hat{r}(\mathbf{x})$ , where

$$\hat{r}(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n \left\| \mathbf{A}_i \mathbf{x}_i + \sum_{j \neq i} \mathbf{A}_j \mathbf{y}_j - \mathbf{b} \right\|^2 + \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{G}_i}^2 + \frac{1-n}{2} \|\mathbf{Ay} - \mathbf{b}\|^2, \quad (21)$$

for any  $\mathbf{y} = [\mathbf{y}_1; \dots; \mathbf{y}_n]$  and  $\mathbf{G}_i \succeq \mathbf{L}'_i - \mathbf{A}_i^\top \mathbf{A}_i$ .

- (3) If  $\mathbf{G}_i = \eta_i \mathbf{I} - \mathbf{A}_i^\top \mathbf{A}_i$  with  $\eta_i \geq \|\mathbf{L}'_i\|_2$ , (21) reduces to

$$\hat{r}(\mathbf{x}) = \sum_{i=1}^n \left\langle \mathbf{x}_i - \mathbf{y}_i, \mathbf{A}_i^\top (\mathbf{Ay} - \mathbf{b}) \right\rangle + \sum_{i=1}^n \frac{\eta_i}{2} \|\mathbf{x}_i - \mathbf{y}_i\|^2 + \frac{1}{2} \|\mathbf{Ay} - \mathbf{b}\|^2. \quad (22)$$

To guarantee that  $\hat{r} \geq r$ , it is required to choose  $\mathbf{L}'_i$  with  $\|\mathbf{L}'_i\|_2$  sufficiently large. Without any additional assumption on  $\mathbf{A}$ , we can choose  $\mathbf{L}'_i = n\mathbf{A}_i^\top \mathbf{A}_i$ . This explains the choice of  $\eta_i > \|\mathbf{L}'_i\|_2 = n\|\mathbf{A}_i\|_2^2$  in L-ADMM-PS (13). However, such a choice of  $\mathbf{L}'_i$  may not be good since it does not make fully use of the structure of  $\mathbf{A}$ , and thus  $\hat{r}$  may not be a tight surrogate of  $r$ . For example, let  $\mathbf{A}_1 = [\mathbf{C}_1; \mathbf{0}]$ ,  $\mathbf{A}_2 = [\mathbf{C}_2; \mathbf{0}]$ ,  $\mathbf{A}_3 = [\mathbf{0}; \mathbf{C}_3]$ ,  $\mathbf{A}_4 = [\mathbf{0}; \mathbf{C}_4]$ , and  $\mathbf{b} = [\mathbf{b}_1; \mathbf{b}_2]$  of compatible sizes. Then  $r(\mathbf{x}) = \frac{1}{2} \|\sum_{i=1}^2 \mathbf{C}_i \mathbf{x}_i - \mathbf{b}_1\|^2 + \frac{1}{2} \|\sum_{i=3}^4 \mathbf{C}_i \mathbf{x}_i - \mathbf{b}_2\|^2$ . We can choose  $\mathbf{L}'_i = 2\mathbf{A}_i^\top \mathbf{A}_i$ , which is much better than  $4\mathbf{A}_i^\top \mathbf{A}_i$ . Actually, the choice of  $\mathbf{L}'_i$  depends on the separability of  $r$ . In practice, it is easy to compute  $\mathbf{L}'_i$  when given  $\mathbf{A}$ . A good choice of  $\mathbf{L}'_i$  gives a tight surrogate  $\hat{r}$ , and this may significantly improve the efficiency of Jacobian ADMMs (see Section 4).

### 3 UNIFIED GAUSS-SEIDEL ADMMs

In this section, we consider solving (1) with  $n = 2$  blocks by a unified framework of Gauss-Seidel ADMMs. In the  $(k+1)$ -th iteration, we compute the majorant surrogate  $\hat{f}^k$  of  $f$  near  $\mathbf{x}^k$ , i.e.,  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^2}(f, \mathbf{x}^k)$  and  $\hat{f}^k$  is separable, i.e.,  $\hat{f}^k(\mathbf{x}) = \hat{f}_1^k(\mathbf{x}_1) + \hat{f}_2^k(\mathbf{x}_2)$ . For  $r_1^k$  and  $r_2^k$  in (8) and (9), we construct their proximal surrogates respectively as follows<sup>1</sup>

$$\hat{r}_1^k(\mathbf{x}_1) = r_1^k(\mathbf{x}_1) + \frac{\beta^{(k)}}{2} \|\mathbf{x}_1 - \mathbf{x}_1^k\|_{\mathbf{G}_1}^2, \quad (23)$$

$$\hat{r}_2^k(\mathbf{x}_2) = r_2^k(\mathbf{x}_2) + \frac{\beta^{(k)}}{2} \|\mathbf{x}_2 - \mathbf{x}_2^k\|_{\mathbf{G}_2}^2, \quad (24)$$

where  $\mathbf{G}_1 \succeq \mathbf{0}$  and  $\mathbf{G}_2 \succ \mathbf{0}$ . Then we update  $\mathbf{x}_1$  and  $\mathbf{x}_2$  by

$$\mathbf{x}_1^{k+1} = \arg \min_{\mathbf{x}_1} \hat{f}_1^k(\mathbf{x}_1) + \hat{r}_1^k(\mathbf{x}_1), \quad (25)$$

$$\mathbf{x}_2^{k+1} = \arg \min_{\mathbf{x}_2} \hat{f}_2^k(\mathbf{x}_2) + \hat{r}_2^k(\mathbf{x}_2). \quad (26)$$

Finally,  $\boldsymbol{\lambda}$  is updated by (4). This leads to the unified framework of Gauss-Seidel ADMMs, as shown in Algorithm 1.

Note that in Algorithm 1,  $f$  is not necessarily separable. In this case, our algorithm and the convergence guarantee shown later are completely new. If  $f$  is already separable, then the objectives in (25) and (26) are majorant surrogates of the ones in (6) and

1. Note that the definitions of  $\hat{r}_i^k$  in Section 3, 4 and 5 are different.

**Algorithm 1** A Unified Framework of Gauss-Seidel ADMMs**For**  $k = 0, 1, 2, \dots$  **do**

- 1) Compute a majorant first-order surrogate  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \mathbf{x}^k)$  with  $\hat{f}^k(\mathbf{x}) = \hat{f}_1^k(\mathbf{x}_1) + \hat{f}_2^k(\mathbf{x}_2)$ .
- 2) Update  $\mathbf{x}_1$  by solving (25).
- 3) Update  $\mathbf{x}_2$  by solving (26).
- 4) Update  $\boldsymbol{\lambda}$  by  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta^{(k)}(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})$ .
- 5) Choose  $\beta^{(k+1)} \geq \beta^{(k)}$ .

**end**

(7), respectively. Many previous Gauss-Seidel ADMMs are special cases by using different majorant surrogates  $\hat{f}_1$  and  $\hat{r}_1^k$  (depending on  $\mathbf{G}_1^k$ ) in Algorithm 1. See Table 1 for a summary.

Assume that there exists an KKT point  $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$  of (1), i.e.,  $\mathbf{A}\mathbf{x}^* = \mathbf{b}$  and  $-\mathbf{A}^\top \boldsymbol{\lambda}^* \in \partial f(\mathbf{x}^*)$ . Previous works prove that ADMMs converge to the KKT point at the rate  $O(1/K)$  ( $K$  is the number of iterations) in different ways. The works [12], [34] give the same rate of ADMM, L-ADMM, and P-ADMM. But they require that both the primal and dual feasible sets should be bounded. The work [22] removes the above assumptions and shows that the convergence rates of L-ADMM-PS and PL-ADMM-PS are

$$f(\bar{\mathbf{x}}^K) - f(\mathbf{x}^*) + \langle \mathbf{A}^\top \boldsymbol{\lambda}^*, \bar{\mathbf{x}}^K - \mathbf{x}^* \rangle + \frac{\alpha}{2} \|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\|^2 \leq O(1/K), \quad (27)$$

where  $\bar{\mathbf{x}}^K$  is a weighted sum of  $\mathbf{x}^k$ 's and  $\alpha > 0$ . Now we give the convergence bound of Algorithm 1 as (27).

**Theorem 1.** In Algorithm 1, assume that  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \mathbf{x}^k)$  with  $\mathbf{P}_i \succeq \mathbf{L}_i \succeq \mathbf{0}$ ,  $i = 1, 2$ ,  $\mathbf{G}_1 \succeq \mathbf{0}$  in (23), and  $\mathbf{G}_2 \succ \mathbf{0}$  in (24). For any  $K > 0$ , let  $\bar{\mathbf{x}}^K = \sum_{k=0}^K \gamma^{(k)} \mathbf{x}^{k+1}$  with  $\gamma^{(k)} = (\beta^{(k)})^{-1} / \sum_{k=0}^K (\beta^{(k)})^{-1}$ . Then

$$f(\bar{\mathbf{x}}^K) - f(\mathbf{x}^*) + \langle \mathbf{A}^\top \boldsymbol{\lambda}^*, \bar{\mathbf{x}}^K - \mathbf{x}^* \rangle + \frac{\beta^{(0)}\alpha}{2} \|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\|^2 \leq \frac{\sum_{i=1}^2 \|\mathbf{x}_i^* - \mathbf{x}_i^0\|_{\mathbf{H}_i^0}^2 + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\|_{\mathbf{H}_3^0}^2}{2 \sum_{k=0}^K (\beta^{(k)})^{-1}}, \quad (28)$$

where  $\alpha = \min \left\{ \frac{1}{2}, \frac{\sigma_{\min}^2(\mathbf{G}_2)}{2\|\mathbf{A}_2\|_2^2} \right\}$ ,  $\mathbf{H}_1^0 = \frac{1}{\beta^{(0)}} \mathbf{L}_1 + \mathbf{G}_1$ ,  $\mathbf{H}_2^0 = \frac{1}{\beta^{(0)}} \mathbf{L}_2 + \mathbf{A}_2^\top \mathbf{A}_2 + \mathbf{G}_2$ , and  $\mathbf{H}_3^0 = \left(1/\beta^{(0)}\right)^2 \mathbf{I}$ .

Consider  $\mathbf{H}_i^0$ ,  $i = 1, 2$ , at the RHS of (28), it can be seen that they depend on  $\mathbf{L}_i$  and  $\mathbf{G}_i$ , which control the difference  $\hat{f} - f$  and  $\hat{r}_i^k - r_i^k$ , respectively. This suggests a faster convergence when using tighter majorant surrogates, though the convergence rate of Gauss-Seidel ADMMs in Algorithm 1 is  $O(1/K)$  when  $\beta^{(k)}$ 's are bounded.

Note that the assumption  $\mathbf{G}_2 \succ \mathbf{0}$  guarantees that  $\alpha > 0$ . Such an assumption is also used in [12], [34] which prove the same convergence rate in different ways. It suggests that using  $\mathbf{G}_2 \succ \mathbf{0}$  instead of  $\mathbf{G}_2 = \mathbf{0}$  in the traditional ADMM can achieve the  $O(1/K)$  convergence rate.

#### 4 UNIFIED JACOBIAN ADMMs

In this section, we consider solving (1) with  $n > 2$  by a unified framework of Jacobian ADMMs. The motivation is to solve (2) inexactly by minimizing a majorant surrogate of  $f(\mathbf{x}) + r^k(\mathbf{x})$ . In the  $(k+1)$ -th iteration, we first compute the majorant surrogate

TABLE 1: Previous Gauss-Seidel ADMMs are special cases of Algorithm 1 with different  $\hat{f}_1$  and  $\mathbf{G}_1$ . In this table,  $\eta_1 > \|\mathbf{A}_1\|_2^2$ .

	$\hat{f}_1^k(\mathbf{x}_1)$	$\mathbf{G}_1$
ADMM	$f_1(\mathbf{x}_1)$	$\mathbf{0}$
P-ADMM	Lipschitz Gradient Surrogate or Proximal Gradient Surrogate	$\mathbf{0}$
L-ADMM	$f_1(\mathbf{x}_1)$	$\eta_1 \mathbf{I} - \mathbf{A}_1^\top \mathbf{A}_1$
PL-ADMM	Lipschitz Gradient Surrogate or Proximal Gradient Surrogate	$\eta_1 \mathbf{I} - \mathbf{A}_1^\top \mathbf{A}_1$

of  $f$  near  $\mathbf{x}^k$ , i.e.,  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \mathbf{x}^k)$ , and  $\hat{f}^k$  is separable,  $\hat{f}^k(\mathbf{x}) = \sum_{i=1}^n \hat{f}_i^k(\mathbf{x}_i)$ . Assume that  $\frac{1}{2}\|\mathbf{A}\mathbf{x}\|^2$  is  $\{\mathbf{L}'_i\}_{i=1}^n$ -smooth. For  $r^k$  in (2), we define its majorant surrogate  $\hat{r}^k$  by using (21), i.e.,  $\hat{r}^k(\mathbf{x}) = \sum_{i=1}^n \hat{r}_i^k(\mathbf{x}_i)$ , where

$$\frac{\hat{r}_i^k(\mathbf{x}_i)}{\beta^{(k)}} = \frac{1}{2} \left\| \mathbf{A}_i \mathbf{x}_i + \sum_{j \neq i} \mathbf{A}_j \mathbf{x}_j^k - \mathbf{b} + \frac{\boldsymbol{\lambda}^k}{\beta^{(k)}} \right\|^2 + \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_i^k\|_{\mathbf{G}_i}^2 + c_i^k, \quad (29)$$

with  $\mathbf{G}_i \succ \mathbf{L}'_i - \mathbf{A}_i^\top \mathbf{A}_i$  and  $c_i^k$ 's are constants satisfying  $\sum_{i=1}^n c_i^k = \frac{1-n}{2} \|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|^2$ . Thus  $\hat{f}^k(\mathbf{x}) + \hat{r}^k(\mathbf{x})$  is a majorant surrogate of  $f(\mathbf{x}) + r^k(\mathbf{x})$  in (2). Now we minimize  $\hat{f}^k(\mathbf{x}) + \hat{r}^k(\mathbf{x})$  instead to update  $\mathbf{x}$ , i.e.,

$$\mathbf{x}^{k+1} = \arg \min_{\mathbf{x}} \hat{f}^k(\mathbf{x}) + \hat{r}^k(\mathbf{x}). \quad (30)$$

Note that both  $\hat{f}$  and  $\hat{r}^k$  are separable. Thus solving (30) is equivalent to updating each  $\mathbf{x}_i$  in parallel, i.e.,

$$\mathbf{x}_i^{k+1} = \arg \min_{\mathbf{x}_i} \hat{f}_i^k(\mathbf{x}_i) + \hat{r}_i^k(\mathbf{x}_i). \quad (31)$$

Finally  $\boldsymbol{\lambda}$  is updated by (4). This leads to the unified framework of Jacobian ADMMs, as shown in Algorithm 2.

If  $f$  is non-separable, then our algorithm and convergence guarantee shown later are completely new. If  $f$  is separable, several previous Jacobian ADMMs are special cases by using different majorant surrogates  $\hat{f}_i$  and  $\hat{r}_i^k$  (depending on  $\mathbf{G}_i$ ) in Algorithm 2. See Table 2 for a summary.

**Theorem 2.** In Algorithm 2, assume that  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \mathbf{x}^k)$  with  $\mathbf{P}_i \succeq \mathbf{L}_i \succeq \mathbf{0}$ ,  $\frac{1}{2}\|\mathbf{A}\mathbf{x}\|^2$  is  $\{\mathbf{L}'_i\}_{i=1}^n$ -smooth, and  $\mathbf{G}_i \succ \mathbf{L}'_i - \mathbf{A}_i^\top \mathbf{A}_i$  in (29). For any  $K > 0$ , let  $\bar{\mathbf{x}}^K = \sum_{k=0}^K \gamma^{(k)} \mathbf{x}^{k+1}$  with  $\gamma^{(k)} = (\beta^{(k)})^{-1} / \sum_{k=0}^K (\beta^{(k)})^{-1}$ . Then

$$f(\bar{\mathbf{x}}^K) - f(\mathbf{x}^*) + \langle \mathbf{A}^\top \boldsymbol{\lambda}^*, \bar{\mathbf{x}}^K - \mathbf{x}^* \rangle + \frac{\beta^{(0)}\alpha}{2} \|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\|^2 \leq \frac{\sum_{i=1}^n \|\mathbf{x}_i^* - \mathbf{x}_i^0\|_{\mathbf{H}_i^0}^2 + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\|_{\mathbf{H}_{n+1}^0}^2}{2 \sum_{k=0}^K (\beta^{(k)})^{-1}}, \quad (32)$$

where  $\alpha = \min \left\{ \frac{1}{2}, \frac{\sigma_{\min}^2(\text{Diag}\{\mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i, i=1, \dots, n\} - \mathbf{A}^\top \mathbf{A})}{2\|\mathbf{A}\|_2^2} \right\}$ ,  $\mathbf{H}_i^0 = \frac{1}{\beta^{(0)}} \mathbf{L}_i + \mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i$ ,  $i = 1, \dots, n$ , and  $\mathbf{H}_{n+1}^0 = \left(1/\beta^{(0)}\right)^2 \mathbf{I}$ .

**Algorithm 2** A Unified Framework of Jacobian ADMMs**For**  $k = 0, 1, 2, \dots$  **do**

- 1) Compute a majorant first-order surrogate  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \mathbf{x}^k)$  with  $\hat{f}^k(\mathbf{x}) = \sum_{i=1}^n \hat{f}_i^k(\mathbf{x}_i)$ .
- 2) Update  $\mathbf{x}_i, i = 1, \dots, n$ , in parallel by solving (31).
- 3) Update  $\boldsymbol{\lambda}$  by  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta^{(k)}(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})$ .
- 4) Choose  $\beta^{(k+1)} \geq \beta^{(k)}$ .

**end**

The above bound implies an interesting connection between the convergence speed and the tightness of the majorant surrogates. For simplicity, let  $\beta^{(k)} = \beta$ . Then (32) reduces to

$$\frac{\sum_{i=1}^n \|\mathbf{x}_i^* - \mathbf{x}_i^0\|_{\beta \mathbf{H}_i^0}^2 + \frac{1}{\beta} \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\|^2}{2(K+1)} \leq \frac{\frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i^* - \mathbf{x}_i^0\|_{\mathbf{L}_i + \beta \mathbf{L}_i'}^2 + \frac{1}{2\beta} \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\|^2}{(K+1)}, \quad (33)$$

where (33) uses  $\mathbf{G}_i \succ \mathbf{L}_i' - \mathbf{A}_i^\top \mathbf{A}_i$ . Now consider the two constant terms in the numerator of (33). The first term controls the tightness of the used majorant surrogate for the  $\mathbf{x}$  updating, i.e.,  $|f^0(\mathbf{x}^*) - \hat{f}^0(\mathbf{x}^*) - f(\mathbf{x}^*) - r(\mathbf{x}^*)| \leq \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i^* - \mathbf{x}_i^0\|_{\mathbf{L}_i + \beta \mathbf{L}_i'}^2$ , which uses (18) with  $\mathbf{x} = \mathbf{x}^*$  and  $k = 0$ . The second term is actually the difference function  $\frac{1}{2\beta} \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^k\|^2$  between  $-\mathcal{L}(\mathbf{x}^{k+1}, \boldsymbol{\lambda}, \beta^{(k)})$  and its majorant surrogate in (16) when  $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$  and  $k = 0$ . So the convergence bound depends on the tightness of the used majorant surrogates for both the primal and dual variables updating. If  $\hat{f}^k + \hat{r}^k$  is tighter (associated to the  $\mathbf{x}$  updating) or  $\beta$  is larger (associated to the  $\boldsymbol{\lambda}$  updating), the algorithm converges faster. In practice, ADMMs stop based on certain criteria induced by the KKT conditions. If  $\beta$  is relatively larger, the algorithm seems to converge faster but the objective function value may be larger. How to choose the best  $\beta$  or  $\beta^{(k)}$  is still an open issue. In this work, we focus the discussion on how to improve the tightness of the majorant surrogate for the primal variable updating.

Note that Algorithm 2 improves previous Jacobian ADMMs which use  $\mathbf{L}_i' = n\mathbf{A}_i^\top \mathbf{A}_i$ . Such a choice of  $\mathbf{L}_i'$  does not fully use the structure of  $\mathbf{A}$  or  $r(\mathbf{x})$  (see the discussions after Lemma 3). Our Algorithm 2 instead uses the  $\{\mathbf{L}_i'\}_{i=1}^n$ -smooth property of  $r(\mathbf{x})$ . This may make the surrogate  $\hat{r}^k(\mathbf{x})$  tighter and thus the algorithm converges faster. In Section 5, we discuss how to further improve the tightness of  $\hat{r}^k(\mathbf{x})$  by introducing alternating minimization in Jacobian ADMMs and the backtracking technique.

**5 MIXED GAUSS-SEIDEL AND JACOBIAN ADMM**

Consider solving (1) with  $n = 2$  by Gauss-Seidel ADMMs and Jacobian ADMMs, the former one will converge faster. The reason is that Jacobian ADMMs require  $\mathbf{G}_i \succ \mathbf{A}_i^\top \mathbf{A}_i$ , while Gauss-Seidel ADMMs only require  $\mathbf{G}_i \succ \mathbf{0}$ . Thus the bound in (28) is expected to be tighter than the one in (32). The superiority of Gauss-Seidel ADMMs over Jacobian ADMMs is that the former first use alternating minimization to reduce the complexity of the problem (fewer variables) and then the used majorant surrogate can be tighter when using majorization minimization.

In this section, we consider problem (1) with  $n > 2$  blocks. We propose the Mixed Gauss-Seidel and Jacobian ADMM (M-ADMM), which introduces the alternating minimization before using majorization minimization. M-ADMM first divides these  $n$

TABLE 2: Previous Jacobian ADMMs are special cases of Algorithm 2 with different  $\hat{f}_i$  and  $\mathbf{G}_i$ . In this table,  $\eta_i > n\|\mathbf{A}_i\|_2^2$ .

	$\hat{f}_i^k(\mathbf{x}_i)$	$\mathbf{G}_i$
L-ADMM-PS	$f_i(\mathbf{x}_i)$	$\eta_i \mathbf{I} - \mathbf{A}_i^\top \mathbf{A}_i$
PL-ADMM-PS	Proximal Gradient Surrogate	$\eta_i \mathbf{I} - \mathbf{A}_i^\top \mathbf{A}_i$
GL-ADMM-PS	$f_i(\mathbf{x}_i)$	$\succ (n-1)\mathbf{A}_i^\top \mathbf{A}_i$

blocks  $\mathbf{x} = [\mathbf{x}_1; \dots; \mathbf{x}_n]$  into two super blocks, i.e.,  $\mathbf{x}_{B_1} = [\mathbf{x}_i, i \in B_1]$  with  $n_1$  blocks of variables, and  $\mathbf{x}_{B_2} = [\mathbf{x}_i, i \in B_2]$  with  $n_2$  blocks of variables, where  $B_1$  and  $B_2$  satisfy  $B_1 \cap B_2 = \emptyset$  and  $B_1 \cup B_2 = \{1, \dots, n\}$ . Then  $\mathbf{x}_{B_1}$  and  $\mathbf{x}_{B_2}$  are updated in a sequential way as Gauss-Seidel ADMMs, while  $\mathbf{x}_i$ 's in each super block are updated in a parallel way as Jacobian ADMMs. As shown later, M-ADMM owns a tighter bound than (32), and thus it will be faster than Jacobian ADMMs. In the following, we first introduce M-ADMM, and then discuss the variable partition and backtracking technique which are crucial for the efficient implementation of M-ADMM in practice.

**5.1 M-ADMM**

Assume that we are given a partition of  $n$  blocks, denoted as  $\{B_1, B_2\}$ . We accordingly partition  $\mathbf{A}$  into  $\mathbf{A}_{B_1} = [\mathbf{A}_i, i \in B_1]$  and  $\mathbf{A}_{B_2} = [\mathbf{A}_i, i \in B_2]$ . Then (1) is equivalent to

$$\min_{\mathbf{x}_{B_1}, \mathbf{x}_{B_2}} f(\mathbf{x}), \text{ s.t. } \mathbf{A}_{B_1} \mathbf{x}_{B_1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2} = \mathbf{b}. \quad (34)$$

In the  $(k+1)$ -th iteration, we first compute the majorant surrogate of  $f$  near  $\mathbf{x}^k$ , i.e.,  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \mathbf{x}^k)$ , and  $\hat{f}^k$  is separable,  $\hat{f}^k(\mathbf{x}) = \hat{f}_{B_1}^k(\mathbf{x}_{B_1}) + \hat{f}_{B_2}^k(\mathbf{x}_{B_2})$ , where  $\hat{f}_{B_i}^k(\mathbf{x}_{B_i}) = \sum_{j \in B_i} \hat{f}_j^k(\mathbf{x}_j)$ ,  $i = 1, 2$ . Then (34) can be solved by updating  $\mathbf{x}_{B_1}$  and  $\mathbf{x}_{B_2}$  as the traditional ADMM, i.e.,

$$\mathbf{x}_{B_1}^{k+1} = \underset{\mathbf{x}_{B_1}}{\operatorname{argmin}} \hat{f}_{B_1}^k(\mathbf{x}_{B_1}) + r_{B_1}^k(\mathbf{x}_{B_1}), \quad (35)$$

$$\mathbf{x}_{B_2}^{k+1} = \underset{\mathbf{x}_{B_2}}{\operatorname{argmin}} \hat{f}_{B_2}^k(\mathbf{x}_{B_2}) + r_{B_2}^k(\mathbf{x}_{B_2}), \quad (36)$$

where

$$r_{B_1}^k(\mathbf{x}_{B_1}) = \frac{\beta^{(k)}}{2} \left\| \mathbf{A}_{B_1} \mathbf{x}_{B_1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^k - \mathbf{b} + \frac{\boldsymbol{\lambda}^k}{\beta^{(k)}} \right\|^2,$$

and

$$r_{B_2}^k(\mathbf{x}_{B_2}) = \frac{\beta^{(k)}}{2} \left\| \mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2} - \mathbf{b} + \frac{\boldsymbol{\lambda}^k}{\beta^{(k)}} \right\|^2.$$

However, the above problems are expensive to solve since they may not be separable w.r.t.  $\mathbf{x}_i$ 's in  $B_1$  or  $B_2$ . Assume that  $\frac{1}{2} \|\mathbf{A}_{B_1} \mathbf{x}_{B_1}\|^2$  is  $\{\mathbf{L}_i'\}_{i \in B_1}$ -smooth. By (21), we construct a majorant surrogate  $\hat{r}_{B_1}^k$  of  $r_{B_1}^k$  near  $\mathbf{x}_{B_1}^k$ , i.e.,  $\hat{r}_{B_1}^k(\mathbf{x}_{B_1}) = \sum_{i \in B_1} \hat{r}_i^k(\mathbf{x}_i)$ , where

$$\begin{aligned} \hat{r}_i^k(\mathbf{x}_i) &= \frac{1}{2} \left\| \mathbf{A}_i \mathbf{x}_i + \sum_{\substack{j \in B_1 \\ j \neq i}} \mathbf{A}_j \mathbf{x}_j^k + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^k - \mathbf{b} + \frac{\boldsymbol{\lambda}^k}{\beta^{(k)}} \right\|^2 \\ &\quad + \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_i^k\|_{\mathbf{G}_i}^2 + c_i^k, \quad i \in B_1, \end{aligned} \quad (37)$$

with  $\mathbf{G}_i \succeq \mathbf{L}_i' - \mathbf{A}_i^\top \mathbf{A}_i$ ,  $i \in B_1$ , and  $c_i^k$ 's satisfying  $\sum_{i \in B_1} c_i^k = \frac{1-n_1}{2} \left\| \mathbf{A} \mathbf{x}^k - \mathbf{b} + \frac{\boldsymbol{\lambda}^k}{\beta^{(k)}} \right\|^2$ . Similarly, assume that  $\frac{1}{2} \|\mathbf{A}_{B_2} \mathbf{x}_{B_2}\|^2$

**Algorithm 3** Mixed Gauss-Seidel and Jacobian ADMM**For**  $k = 0, 1, 2, \dots$  **do**

- 1) Compute a majorant first-order surrogate  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \mathbf{x}^k)$  with  $\hat{f}^k(\mathbf{x}) = \sum_{i=1}^n \hat{f}_i^k(\mathbf{x}_i)$ .
- 2) For all  $i \in B_1$ , update  $\mathbf{x}_i$ 's in parallel by solving (39).
- 3) For all  $i \in B_2$ , update  $\mathbf{x}_i$ 's in parallel by solving (40).
- 4) Update  $\boldsymbol{\lambda}$  by  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta^{(k)}(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})$ .
- 5) Choose  $\beta^{(k+1)} \geq \beta^{(k)}$ .

**end**

is  $\{\mathbf{L}'_i\}_{i \in B_2}$ -smooth. Then a majorant surrogate  $\hat{r}_{B_2}^k$  of  $r_{B_2}^k$  near  $\mathbf{x}_{B_2}^k$  is  $\hat{r}_{B_2}^k(\mathbf{x}_{B_2}) = \sum_{i \in B_2} \hat{r}_i^k(\mathbf{x}_i)$ , where

$$\frac{\hat{r}_i^k(\mathbf{x}_i)}{\beta^{(k)}} = \frac{1}{2} \left\| \mathbf{A}_i \mathbf{x}_i + \sum_{\substack{j \in B_2 \\ j \neq i}} \mathbf{A}_j \mathbf{x}_j^k + \mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} - \mathbf{b} + \frac{\boldsymbol{\lambda}^k}{\beta^{(k)}} \right\|^2 + \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_i^k\|_{\mathbf{G}_i}^2 + c_i^k, \quad i \in B_2, \quad (38)$$

with  $\mathbf{G}_i \succ \mathbf{L}'_i - \mathbf{A}_i^\top \mathbf{A}_i$ ,  $i \in B_2$ , and  $c_i^k$ 's satisfying  $\sum_{i \in B_2} c_i^k = \frac{1-n_2}{2} \left\| \mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^k - \mathbf{b} + \frac{\boldsymbol{\lambda}^k}{\beta^{(k)}} \right\|^2$ . By replacing  $r_{B_1}^k(\mathbf{x}_{B_1})$  and  $r_{B_2}^k(\mathbf{x}_{B_2})$  with their majorant surrogates  $\hat{r}_{B_1}^k(\mathbf{x}_{B_1})$  and  $\hat{r}_{B_2}^k(\mathbf{x}_{B_2})$  in (35) and (36) respectively, we update  $\mathbf{x}_{B_1}$  and  $\mathbf{x}_{B_2}$  by

$$\mathbf{x}_{B_1}^{k+1} = \underset{\mathbf{x}_{B_1}}{\operatorname{argmin}} \hat{f}_{B_1}^k(\mathbf{x}_{B_1}) + \hat{r}_{B_1}^k(\mathbf{x}_{B_1}),$$

$$\mathbf{x}_{B_2}^{k+1} = \underset{\mathbf{x}_{B_2}}{\operatorname{argmin}} \hat{f}_{B_2}^k(\mathbf{x}_{B_2}) + \hat{r}_{B_2}^k(\mathbf{x}_{B_2}).$$

Note that the above two problems are separable for each  $\mathbf{x}_i$  in  $B_1$  and  $B_2$ . They are respectively equivalent to

$$\mathbf{x}_i^{k+1} = \underset{\mathbf{x}_i}{\operatorname{argmin}} \hat{f}_i^k(\mathbf{x}_i) + \hat{r}_i^k(\mathbf{x}_i), \quad i \in B_1, \quad (39)$$

$$\mathbf{x}_i^{k+1} = \underset{\mathbf{x}_i}{\operatorname{argmin}} \hat{f}_i^k(\mathbf{x}_i) + \hat{r}_i^k(\mathbf{x}_i), \quad i \in B_2. \quad (40)$$

Finally  $\boldsymbol{\lambda}$  is updated by (4). This leads to the Mixed Gauss-Seidel and Jacobian ADMM (M-ADMM), as shown in Algorithm 3. Now we give its convergence bound as (27).

**Theorem 3.** In Algorithm 3, assume that  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \mathbf{x}^k)$  with  $\mathbf{P}_i \succeq \mathbf{L}_i \succeq \mathbf{0}$ ,  $\frac{1}{2} \|\mathbf{A}_{B_1} \mathbf{x}_{B_1}\|^2$  is  $\{\mathbf{L}'_i\}_{i \in B_1}$ -smooth,  $\frac{1}{2} \|\mathbf{A}_{B_2} \mathbf{x}_{B_2}\|^2$  is  $\{\mathbf{L}'_i\}_{i \in B_2}$ -smooth,  $\mathbf{G}_i \succeq \mathbf{L}'_i - \mathbf{A}_i^\top \mathbf{A}_i$ ,  $i \in B_1$  in (37) and  $\mathbf{G}_i \succ \mathbf{L}'_i - \mathbf{A}_i^\top \mathbf{A}_i$ ,  $i \in B_2$  in (38). For any  $K > 0$ , let  $\bar{\mathbf{x}}^K = \sum_{k=0}^K \gamma^{(k)} \mathbf{x}^{k+1}$  with  $\gamma^{(k)} = (\beta^{(k)})^{-1} / \sum_{k=0}^K (\beta^{(k)})^{-1}$ . Then

$$f(\bar{\mathbf{x}}^K) - f(\mathbf{x}^*) + \langle \mathbf{A}^\top \boldsymbol{\lambda}^*, \bar{\mathbf{x}}^K - \mathbf{x}^* \rangle + \frac{\beta^{(0)} \alpha}{2} \|\mathbf{A} \bar{\mathbf{x}}^K - \mathbf{b}\|^2 \leq \frac{\sum_{j=1}^2 \|\mathbf{x}_{B_j}^* - \mathbf{x}_{B_j}^0\|_{\mathbf{H}_j^0}^2 + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\|_{\mathbf{H}_3^0}^2}{2 \sum_{k=0}^K (\beta^{(k)})^{-1}}, \quad (41)$$

where  $\alpha = \min \left\{ \frac{1}{2}, \frac{\sigma_{\min}(\operatorname{Diag}\{\mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i, i \in B_2\} - \mathbf{A}_{B_2}^\top \mathbf{A}_{B_2})}{2 \|\mathbf{A}_{B_2}\|_2^2} \right\}$ ,  $\mathbf{H}_1^0 = \operatorname{Diag} \left\{ \frac{1}{\beta^{(0)}} \mathbf{L}_i + \mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i, i \in B_1 \right\} - \mathbf{A}_{B_1}^\top \mathbf{A}_{B_1}$ ,  $\mathbf{H}_2^0 = \operatorname{Diag} \left\{ \frac{1}{\beta^{(0)}} \mathbf{L}_i + \mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i, i \in B_2 \right\}$ , and  $\mathbf{H}_3^0 = (1/\beta^{(0)})^2 \mathbf{I}$ .

M-ADMM in Algorithm 3 further unifies Gauss-Seidel ADMMs in Algorithm 1 and Jacobian ADMMs in Algorithm 2.

**Algorithm 4** M-ADMM with backtracking**Initialization:**  $k = 0$ ,  $\mathbf{x}_i^k, \mathbf{G}_i^k \succ \mathbf{0}$ ,  $\boldsymbol{\lambda}^k, \beta^k > 0$ ,  $\tau > 0$ ,  $\mu > 1$ .**For**  $k = 0, 1, 2, \dots$  **do**

- 1) Compute a majorant first-order surrogate  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \mathbf{x}^k)$  with  $\hat{f}^k(\mathbf{x}) = \sum_{i=1}^n \hat{f}_i^k(\mathbf{x}_i)$ .
- 2) Set  $\mathbf{G}_i = \mathbf{G}_i^k$  and compute  $\mathbf{x}_i^{k+1}$  by (39)-(40).
- 3) If (42) does not hold, set  $\mathbf{G}_i^k = \mu \mathbf{G}_i^k$ ,  $i \in B_1$ . Go to 2).  
If (44) does not hold, set  $\mathbf{G}_i^k = \mu \mathbf{G}_i^k$ ,  $i \in B_2$ . Go to 2).
- 4) Update  $\boldsymbol{\lambda}$  by  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta^{(k)}(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})$ .
- 5) Choose  $\beta^{(k+1)} \geq \beta^{(k)}$ . Set  $\mathbf{G}_i^{k+1} = \mathbf{G}_i^k$ ,  $i = 1, \dots, n$ .

**end**

If  $n = 2$ , let  $B_1 = \{1\}$  and  $B_2 = \{2\}$ . Then M-ADMM degenerates into the Gauss-Seidel ADMMs, and (41) reduces to (28). If  $n > 2$ , let  $B_1 = \emptyset$  and  $B_2 = \{1, \dots, n\}$ . Then M-ADMM degenerates into the Jacobian ADMMs, and (41) reduces to (32). More importantly, for the case of  $n > 2$  and other choices of  $B_1$  and  $B_2$ , M-ADMM will be faster than Jacobian ADMMs, since the right hand side of (41) may be much smaller than the one of (32). This is due to their different choices of  $\mathbf{G}_i$ . Without the additional assumption on the structure of  $\mathbf{A}$ , Jacobian ADMMs require  $\mathbf{G}_i \succ (n-1)\mathbf{A}_i^\top \mathbf{A}_i$  for all  $i = 1, \dots, n$ , while M-ADMM only requires  $\mathbf{G}_i \succ (n_1-1)\mathbf{A}_i^\top \mathbf{A}_i$  for  $i \in B_1$  and  $\mathbf{G}_i \succ (n_2-1)\mathbf{A}_i^\top \mathbf{A}_i$  for  $i \in B_2$ . Note that  $n = n_1 + n_2$ . The improvement benefits from the sequential updating rules of  $\mathbf{x}_{B_1}$  and  $\mathbf{x}_{B_2}$  by using tighter majorant surrogates in M-ADMM. Indeed, M-ADMM only needs to majorize  $r_{B_1}^k(\mathbf{x}_{B_1})$  in (35) and  $r_{B_2}^k(\mathbf{x}_{B_2})$  in (36) for  $\mathbf{x}_{B_1}$  and  $\mathbf{x}_{B_2}$  respectively, while Jacobian ADMMs need to majorize  $r^k(\mathbf{x})$  in (3) for all  $\mathbf{x}_i$ 's simultaneously.

Note that the work [13] proposes a block-wise ADMM which is another special case of our Algorithm 3. But their considered problem is more specific and the convergence guarantee requires much stronger assumptions, e.g.,  $\mathbf{A}_i$  that has a full column rank.

**5.2 M-ADMM with Backtracking**

We have given the convergence guarantee of M-ADMM when fixing  $\mathbf{G}_i$ . In practice, we can estimate it by the backtracking technique which will lead to tighter majorant surrogate. The effectiveness has been verified in first-order optimization [1]. Now, we introduce the backtracking technique into M-ADMM.

To guarantee the convergence,  $\mathbf{G}_i$  can be replaced by  $\mathbf{G}_i^k$  such that  $r_{B_1}^k(\mathbf{x}_{B_1}^{k+1}) \leq \hat{r}_{B_1}^k(\mathbf{x}_{B_1}^{k+1})$  and  $r_{B_2}^k(\mathbf{x}_{B_2}^{k+1}) \leq \hat{r}_{B_2}^k(\mathbf{x}_{B_2}^{k+1})$ . They are guaranteed when

$$\|\mathbf{A}_{B_1}(\mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k)\|^2 \leq \sum_{i \in B_1} \|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|_{\mathbf{G}_i^k + \mathbf{A}_i^\top \mathbf{A}_i}^2, \quad (42)$$

$$\|\mathbf{A}_{B_2}(\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k)\|^2 \leq \sum_{i \in B_2} \|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|_{\mathbf{G}_i^k + \mathbf{A}_i^\top \mathbf{A}_i}^2. \quad (43)$$

To achieve the  $O(1/K)$  convergence rate, we replace (43) as

$$\tau \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|^2 \leq \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2^k - \mathbf{A}_{B_2}^\top \mathbf{A}_{B_2}}^2, \quad (44)$$

for some small constant  $\tau > 0$  and  $\mathbf{K}_2^k = \operatorname{Diag}\{\mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^k, i \in B_2\}$ . In this case, we may be able to find  $\mathbf{G}_i^k$  with relatively smaller  $\|\mathbf{G}_i^k\|_2$ , and thus  $\hat{r}_{B_1}^k(\mathbf{x}_{B_1}^{k+1})$  and  $\hat{r}_{B_2}^k(\mathbf{x}_{B_2}^{k+1})$  are tighter upper bounds of  $r_{B_1}^k(\mathbf{x}_{B_1}^{k+1})$  and  $r_{B_2}^k(\mathbf{x}_{B_2}^{k+1})$ , respectively. This leads to a better approximation solution and improves the efficiency. We summarize M-ADMM with backtracking in Algorithm



4. Note that Step 3) will only be performed for finitely many times. Similarly, the convergence guarantee is given as follows.

**Theorem 4.** *In Algorithm 4, assume that  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \mathbf{x}^k)$  with  $\mathbf{P}_i \succeq \mathbf{L}_i \succeq \mathbf{0}$ . Then (41) holds with  $\mathbf{H}_1^0 = \text{Diag} \left\{ \frac{1}{\beta^{(0)}} \mathbf{L}_i + \mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^0, i \in B_1 \right\} - \mathbf{A}_{B_1}^\top \mathbf{A}_{B_1}$ ,  $\mathbf{H}_2^0 = \text{Diag} \left\{ \frac{1}{\beta^{(0)}} \mathbf{L}_i + \mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^0, i \in B_2 \right\}$ ,  $\mathbf{H}_3^0 = \left(1/\beta^{(0)}\right)^2 \mathbf{I}$ , and  $\alpha = \min \left\{ \frac{1}{2}, \frac{\tau}{2\|\mathbf{A}_{B_2}\|_2^2} \right\}$ .*

Note that Algorithm 4 reduces to Algorithm 3 by choosing  $\mathbf{G}_i$ 's in Theorem 3. Theorem 3 is a special case of Theorem 4 by setting  $\tau = \sigma_{\min}^2(\text{Diag} \{ \mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i, i \in B_2 \} - \mathbf{A}_{B_2}^\top \mathbf{A}_{B_2})$ . So we only give the proof of Theorem 4 in Appendix. It is worth mentioning that, when using backtracking,  $\hat{r}_{B_j}^k$  is not a majorant first-order surrogate of  $r_{B_j}^k$ , since the majorization condition may not hold. Actually,  $\hat{r}_{B_j}^k$  only majorizes  $r_{B_j}^k$  locally at  $\mathbf{x}_{B_j}^{k+1}$ . But this is sufficient for the convergence proof, since the formulations of  $r_{B_j}^k$  and  $\hat{r}_{B_j}^k$  are known and we are able to use their specific properties instead of (20) in the proofs.

### 5.3 Variable Partition

For (1) with  $n > 2$ , M-ADMM requires partitioning variables into 2 super blocks  $B_1$  and  $B_2$ . Different variable partitions lead to different choices of  $\mathbf{L}_i^k$  which controls the tightness of the majorant surrogates, and thus the convergence behaviors of M-ADMM are different. Looking for an intelligent way of variable partition may significantly improve the efficiency of M-ADMM. We discuss how to partition variables in three cases by considering the property of  $\mathbf{A}_i$ 's in (1). The principle is to find a partition such that the constructed surrogate  $\hat{r}_{B_1}^k$  for  $r_{B_1}^k$  in (35) and  $\hat{r}_{B_2}^k$  for  $r_{B_2}^k$  in (36) can be as tight as possible.

**Case I (complex case):**  $\mathbf{A}_i^\top \mathbf{A}_l \neq \mathbf{0}$  for any  $i \neq l$ . This case is complex since  $r_{B_j}^k, j = 1, 2$  in (35)-(36) are non-separable for any partition. Then the separable surrogates  $\hat{r}_{B_j}^k$ 's may be loose when considering the choices of  $\mathbf{G}_i$  in (37)-(38). As suggested by Theorem 3, to tighten the bound of (41), a reasonable partition is to make  $L_{B_1} + L_{B_2}$ , where  $L_{B_1} = (n_1 - 1) \sum_{i \in B_1} \|\mathbf{A}_i\|_2^2 - \|\mathbf{A}_{B_1}\|_2^2$  and  $L_{B_2} = (n_2 - 1) \sum_{i \in B_2} \|\mathbf{A}_i\|_2^2$ , as small as possible<sup>2</sup>. We have a heuristic approach to this end: Step 1: Sort  $\|\mathbf{A}_i\|_2^2$ 's in a descending order.

Step 2: Group the largest  $n_1$  elements as the first block and the rest as the second block.

Step 3: The best value of  $n_1$  is the one which minimizes  $L_{B_1} + L_{B_2}$  by a one-shot searching from 1 to  $n$ .

**Case II (simple case):** there exists a partition such that

$$\mathbf{A}_i^\top \mathbf{A}_l = \mathbf{0}, i \neq l, \text{ for any } i, l \in B_1 \text{ and } i, l \in B_2. \quad (45)$$

This case is simple since the above partition makes  $r_{B_j}^k, j = 1, 2$  in (35)-(36) separable. Then  $\hat{r}_{B_j}^k$ 's tend to be tight since we can compute each  $\hat{r}_{B_j}^k$  independently and use  $\mathbf{G}_i \succeq \mathbf{0}, i \in B_1$  in (37) and  $\mathbf{G}_i \succ \mathbf{0}, i \in B_2$  in (38). Even, the per-iteration cost is cheap when using  $\hat{r}_i^k = r_i^k$  for many problems in practice. In this case, (34) can be solved by (35)-(36), which is similar to the standard ADMM. For example, the Low-Rank Representation model in [24] satisfies (45),

$$\min_{\mathbf{Z}, \mathbf{J}, \mathbf{E}} \|\mathbf{J}\|_* + \lambda \|\mathbf{E}\|_{2,1}, \text{ s.t. } \mathbf{X} = \mathbf{AZ} + \mathbf{E}, \mathbf{Z} = \mathbf{J}, \quad (46)$$

2. If  $n_1$  is not very small,  $\|\mathbf{A}_{B_1}\|_2^2$  is usually much smaller than  $(n_1 - 1) \sum_{i \in B_1} \|\mathbf{A}_i\|_2^2$ . We can use  $L_{B_1} = (n_1 - 1) \sum_{i \in B_1} \|\mathbf{A}_i\|_2^2$  in this case.

where  $\lambda > 0$ . The augmented Lagrangian function is

$$\begin{aligned} \mathcal{L}(\mathbf{Z}, \mathbf{J}, \mathbf{E}, \lambda_1, \lambda_2) = & \|\mathbf{J}\|_* + \lambda \|\mathbf{E}\|_{2,1} + \langle \lambda_1, \mathbf{X} - \mathbf{AZ} - \mathbf{E} \rangle \\ & + \langle \lambda_2, \mathbf{Z} - \mathbf{J} \rangle + \frac{\beta}{2} (\|\mathbf{X} - \mathbf{AZ} - \mathbf{E}\|^2 + \|\mathbf{Z} - \mathbf{J}\|^2). \end{aligned}$$

Based on the partition  $\{\mathbf{J}, \mathbf{E}\}$  and  $\{\mathbf{Z}\}$ , they can be updated by

$$\begin{cases} \{\mathbf{J}^{k+1}, \mathbf{E}^{k+1}\} = \arg \min_{\mathbf{J}, \mathbf{E}} \mathcal{L}(\mathbf{Z}^k, \mathbf{J}, \mathbf{E}, \lambda_1^k, \lambda_2^k), \\ \mathbf{Z}^{k+1} = \arg \min_{\mathbf{Z}} \mathcal{L}(\mathbf{Z}, \mathbf{J}^{k+1}, \mathbf{E}^{k+1}, \lambda_1^k, \lambda_2^k). \end{cases}$$

This is the standard ADMM and its convergence is guaranteed. Note that  $\mathcal{L}(\mathbf{Z}^k, \mathbf{J}, \mathbf{E}, \lambda_1^k, \lambda_2^k)$  is separable w.r.t.  $\mathbf{J}$  and  $\mathbf{E}$  and thus  $\mathbf{J}^{k+1}$  and  $\mathbf{E}^{k+1}$  can be computed independently. The updates of the three blocks are similar to the naive multi-block extension of ADMM used in [24], but in different updating orders. Our simple modification fixes the convergence issue of the naive multi-block extension of ADMM in [24] for (46).

In computer vision and signal processing, there are a lot of multi-blocks problems, or their equivalent ones by introducing auxiliary variables, with the property (45) and thus can be solved more efficiently by the Gauss-Seidel ADMMs than Jacobian ADMMs, e.g., sparse subspace clustering model (70) in [8], nonnegative matrix completion problem (143) in [22], multi-task low-rank affinity pursuit model (4) in [5], sparse spectral clustering model (6) in [30], nonnegative low-rank and sparse graph model (5) in [42], simultaneously structured models (3.3) in [35], convex program (8) in [4] for graph clustering, robust multi-view spectral clustering model (3) in [41] and consolidated tensor recovery model (2.6) in [16]. However, some of previous works do not use the property (45) to implement the efficient ADMMs, and this is the reason why we release the toolbox.

**Case III (other cases):** neither assumptions in Case I and Case II holds. It is generally difficult to find the best partition in this case. But one can combine the ideas in both Case I and II. For example, there exists one or more subgroups  $B_S$ , such that  $\mathbf{A}_i^\top \mathbf{A}_l = \mathbf{0}, i \neq l$ , for any  $i, l \in B_S$ . We can put the whole subgroup in one super block, i.e.,  $B_S \subset B_1$ .

In practice, one usually needs to reformulate the original problem as an equivalent one by introducing auxiliary variables such that the subproblem in ADMMs can be simple. When designing efficient ADMMs, the problem reformulation and the above variable partition strategies should be considered simultaneously. Some more examples can be found in our released toolbox.

## 6 EXPERIMENTS

In this section, we conduct several experiments to show the effectiveness of our new ADMMs. All the algorithms are implemented by Matlab and are tested on a PC with 8 GB of RAM and Intel Core 2 Quad CPU Q9550. The details of the compared solvers can be found in the supplementary material.

### 6.1 Experimental Analysis of M-ADMM

Besides the unified analysis of several variants of ADMMs, another main contribution of this work is the proposed M-ADMM for multi-block problems. In this subsection, our purpose is to perform some analyses on M-ADMM. For the simplicity, we first consider the following nonnegative sparse coding problem

$$\min_{\{\mathbf{x}_i\}} \sum_{i=1}^n \|\mathbf{x}_i\|_1, \text{ s.t. } \mathbf{y} = \sum_{i=1}^n \mathbf{A}_i \mathbf{x}_i, \mathbf{x}_i \geq \mathbf{0}, \quad (47)$$



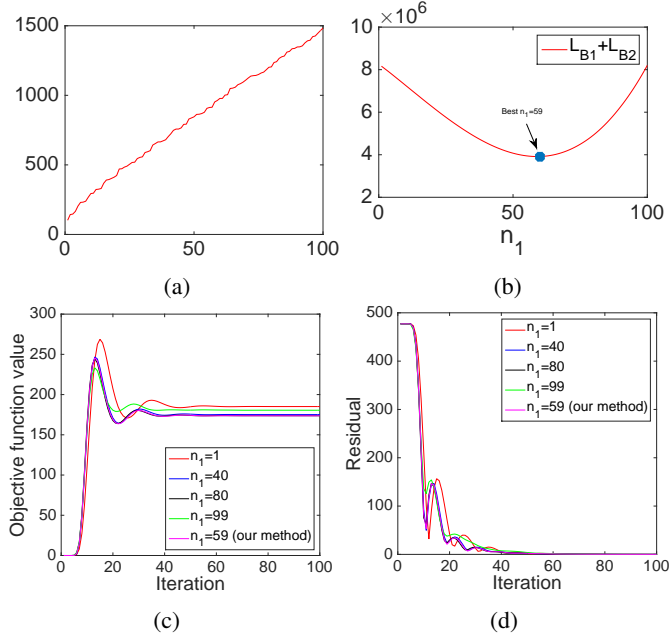


Fig. 1: Plots of (a) sorted  $\{\|\mathbf{A}_i\|_2^2, i = 1, \dots, n\}$ ; (b)  $L_{B_1} + L_{B_2}$  v.s.  $n_1$ ; (c)  $f(\mathbf{x}^k)$  v.s.  $k$  and (d)  $\|\mathbf{Ax}^k - \mathbf{b}\|$  v.s.  $k$  based on different partitions (corresponding to different  $n_1$ ).

### 6.1.1 Analysis of the Proposed Partition Strategy

We conduct an experiment to compare the different convergence behaviors of M-ADMM with different variable partitions and demonstrate the effectiveness of the proposed partition method for Case I in Section 5.3. By choosing  $\mathbf{G}_i \succeq \eta_i \mathbf{I} - \mathbf{A}_i^\top \mathbf{A}_i$  in (37) and (38), M-ADMM solves (47) by the following rules

$$\begin{cases} \mathbf{x}_i^{k+1} = \arg \min_{\mathbf{x}_i \geq 0} \|\mathbf{x}_i\|_1 + \frac{\beta^{(k)} \eta_i}{2} \|\mathbf{x}_i - \mathbf{u}_i^k\|^2, & i \in B_1, \\ \mathbf{x}_i^{k+1} = \arg \min_{\mathbf{x}_i \geq 0} \|\mathbf{x}_i\|_1 + \frac{\beta^{(k)} \eta_i}{2} \|\mathbf{x}_i - \mathbf{v}_i^k\|^2, & i \in B_2, \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta^{(k)} (\mathbf{Ax}^{k+1} - \mathbf{y}), \end{cases}$$

where  $\mathbf{u}_i^k = \mathbf{x}_i^k - \frac{\mathbf{A}_i^\top (\boldsymbol{\lambda}^k + \beta^{(k)} (\mathbf{Ax}^k - \mathbf{y}))}{\beta^{(k)} \eta_i}$ ,  $\mathbf{v}_i^k = \mathbf{x}_i^k - \frac{\mathbf{A}_i^\top (\boldsymbol{\lambda}^k + \beta^{(k)} (\mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^k - \mathbf{y}))}{\beta^{(k)} \eta_i}$ ,  $\eta_i \geq n_1 \|\mathbf{A}_i\|_2^2$ ,  $i \in B_1$ , and  $\eta_i > n_2 \|\mathbf{A}_i\|_2^2$ ,  $i \in B_2$ . In M-ADMM,  $\mathbf{x}_i$  and  $\boldsymbol{\lambda}$  are initialized as zeros. We set  $\beta^{(0)} = 10^{-4}$  and update  $\beta^{(k+1)} = \min(\rho \beta^{(k)}, 10^6)$  with  $\rho = 1.1$ . Let  $\eta_i = n_1 \|\mathbf{A}_i\|_2^2$  for  $i \in B_1$ , and  $\eta_i = 1.02 n_2 \|\mathbf{A}_i\|_2^2$  for  $i \in B_2$ . We test M-ADMM for (47) on the synthetic data generated as follows. We set  $n = 100$ ,  $d = 50$ ,  $m_i = 10i$  and the elements of  $\mathbf{A}_i \in \mathbb{R}^{d \times m_i}$  are independently sampled from an  $N(0, 1)$  distribution. We generate  $\mathbf{x}$  with 90% elements being zeros and others independently sampled from an  $N(0, 1)$  distribution. Then  $\mathbf{y} = [\mathbf{A}_1, \dots, \mathbf{A}_n] \mathbf{x}$ . The sizes of  $\mathbf{A}_i$ 's are different, and so are the Lipschitz constants  $\|\mathbf{A}_i\|_2^2$ 's. We plot the sorted  $\|\mathbf{A}_i\|_2^2$ 's in Figure 1 (a). M-ADMM requires dividing these  $n$  blocks of variables into two super blocks, i.e.,  $\mathbf{x}_{B_1}$  with  $n_1$  blocks, and  $\mathbf{x}_{B_2}$  with  $n_2$  blocks. Our partition strategy finds  $n_1$  by minimizing  $L_{B_1} + L_{B_2}$ , where  $L_{B_1} = (n_1 - 1) \sum_{i \in B_1} \|\mathbf{A}_i\|_2^2 - \|\mathbf{A}_{B_1}\|_2^2$  and  $L_{B_2} = (n_2 - 1) \sum_{i \in B_2} \|\mathbf{A}_i\|_2^2$ . In this experiment, our method gives the best  $n_1 = 59$ . See the plot of  $L_{B_1} + L_{B_2}$  v.s.  $n_1$  in Figure 1 (b). Note that one may have many other choices of  $n_1 \in \{1, 2, \dots, 100\}$ . Figure 1 (c) plots the objective function value  $f(\mathbf{x}^k)$  v.s. iteration  $k$  ( $\leq 100$ ) and Figure 1 (d) plots the

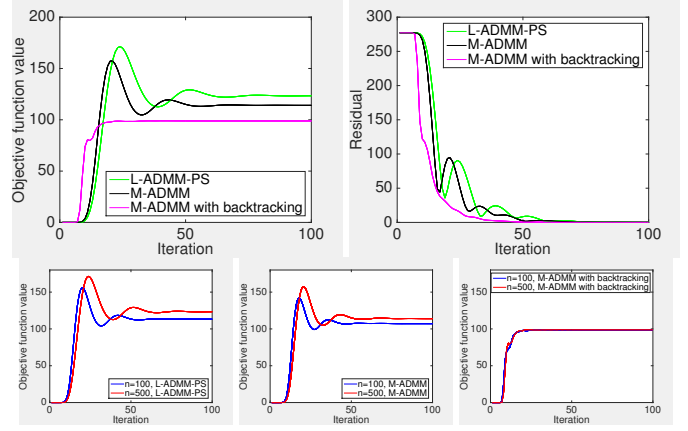


Fig. 2: Top row: comparison of L-ADMM-PS, M-ADMM and M-ADMM with backtracking based on  $f(\mathbf{x}^k)$  v.s.  $k$  (left) and  $\|\mathbf{Ax}^k - \mathbf{b}\|$  v.s.  $k$  (right) in the case  $n = 500$ . Bottom row: comparison of L-ADMM-PS (left), M-ADMM (middle), and M-ADMM with backtracking (right) in different cases of  $n = 100$  and  $n = 500$ .

residual  $\|\mathbf{Ax}^k - \mathbf{b}\|$  v.s. iteration  $k$  ( $\leq 100$ ), based on different choices of  $n_1 \in \{1, 40, 80, 99, 69\}$ . Generally, the convergences of  $\|\mathbf{Ax}^k - \mathbf{b}\|$  based on different partitions are quite similar since it heavily depends on the same updating rule of  $\beta^{(k+1)}$ . However, different  $n_1$  leads to quite different convergences of  $f(\mathbf{x}^k)$ , and  $n_1 = 59$ , predicted by our method, performs well. The choice of  $n_1 = 1$  is the worst case since  $L_{B_1} + L_{B_2}$  is the largest. These results verify that M-ADMM converge faster when using our proposed variable partition strategy, which leads to tight majorant surrogates.

### 6.1.2 Analysis of M-ADMM with backtracking

We conduct three experiments to show the advantage of M-ADMM with backtracking, which uses tighter majorant surrogate, over L-ADMM-PS and M-ADMM. We still consider (47) on synthetic data. We generate  $\mathbf{A} \in \mathbb{R}^{d \times m}$ , where  $d = 50$  and  $m = 10,000$ , with its elements independently sampled from an  $N(0, 1)$  distribution. We generate  $\mathbf{x}$  with 90% elements being zeros and others independently sampled from an  $N(0, 1)$  distribution, and  $\mathbf{y} = \mathbf{Ax}$ . Then we uniformly split  $\mathbf{A}$  and  $\mathbf{x}$  into  $n$  blocks,  $[\mathbf{A}_1, \dots, \mathbf{A}_n]$  and  $\mathbf{x} = [\mathbf{x}_1; \dots; \mathbf{x}_n]$ , respectively. We consider two cases:  $n = 100$  and  $n = 500$ . Though  $n$  is different, the solved problems are equivalent. We are interested in the different convergence behaviors of the used solvers in both cases. In M-ADMM with backtracking, we set  $\tau = 1.3$ ,  $\eta_i = 0.01 n_1 \|\mathbf{A}_i\|_2^2$ ,  $i \in B_1$  and  $\eta_i = 0.01 n_2 \|\mathbf{A}_i\|_2^2$ ,  $i \in B_2$ . The other settings and the initializations are the same as M-ADMM in Section 6.1.1. Note that though the backtracking in Algorithm 4 requires some additional cost to estimate  $\eta_i$ 's, the cost can be ignored since the conditions in (42) and (44) fail only in a few iterations. Considering that the per-iteration complexity of the three solvers are the same, we simply compare their performance based on  $f(\mathbf{x}^k)$  v.s.  $k$  and  $\|\mathbf{Ax}^k - \mathbf{b}\|$  v.s.  $k$ .

Figure 2 shows the comparison results. In Figure 2 (a)-(b), we consider the case  $n = 500$  and compare the three solvers based on  $f(\mathbf{x}^k)$  v.s.  $k$  ( $k \leq 100$ ) and  $\|\mathbf{Ax}^k - \mathbf{b}\|$  v.s.  $k$ . It can be seen that M-ADMM with backtracking achieves the smallest objective function value when the algorithm converges and it reduces the residual much faster than the other two methods. M-ADMM also outperforms L-ADMM-PS. These results well veri-

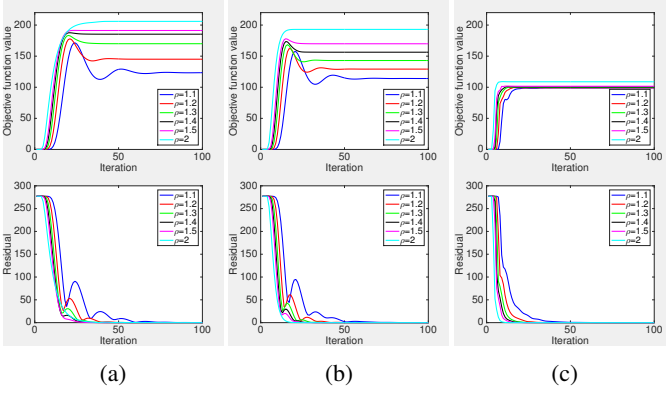


Fig. 3: Comparison of (a) L-ADMM-PS, (b) M-ADMM and (c) M-ADMM with backtracking on different choices of  $\rho$ , where  $\beta^{(k+1)} = \rho\beta^{(k)}$ . Top row: plots of  $f(\mathbf{x}^k)$  v.s.  $k$ ; bottom row: plots of  $\|\mathbf{Ax}^k - \mathbf{b}\|$  v.s.  $k$ .

fies the effectiveness of M-ADMM and the proposed backtracking technique, and are consistent with our theoretical analysis. Second, we compare the convergence behaviors of the three solvers based on different splits of  $\mathbf{A}$ , i.e.,  $n = 100$  and  $n = 500$ . From Figure 2 (c)-(d), it can be seen that L-ADMM-PS and M-ADMM for the case  $n = 100$  perform much better than the case  $n = 500$ , respectively. This is not a surprise since both two solvers use constant  $\eta_i$ 's which depend on the block number (see Theorem 3). Intuitively, the smaller  $n$  leads to a tighter majorant surrogate, e.g., (37), and thus it leads to a better approximated solution. However, M-ADMM with backtracking performs the best and it is not sensitive to  $n$ , since it estimates  $\eta_i$ 's locally and this leads to a tight majorant surrogate.

Furthermore, we compare the three solvers based on different choices of  $\rho \in \{1.1, 1.2, 1.3, 1.4, 1.5, 2\}$ , where  $\beta^{(k+1)} = \rho\beta^{(k)}$ . We test on the same dataset as the above experiment with  $n = 500$ , and plot  $f(\mathbf{x}^k)$  v.s.  $k$  and  $\|\mathbf{Ax}^k - \mathbf{b}\|$  v.s.  $k$  in Figure 3. For all the three solvers, when  $\rho$  is larger, the residual  $\|\mathbf{Ax}^k - \mathbf{b}\|$  decreases faster. More importantly, the price is that the objective  $f(\mathbf{x}^k)$  decreases slower. Considering the convergence of  $f(\mathbf{x}^k)$ , both L-ADMM-PS and M-ADMM are sensitive to the choice of  $\rho$ , though the later one performs better. However, M-ADMM with backtracking performs very well even when  $\rho$  increases. The reason is that the larger  $\rho$  implies that  $\beta^{(k)}$  increases much faster and this makes the majorant surrogates in (37)-(38) much looser. In contrast, the surrogates  $\hat{r}_{B_1}^k$  and  $\hat{r}_{B_2}^k$  in M-ADMM with backtracking are computed locally based on (42) and (44) and thus the surrogates are much tighter. This experiment verifies that the backtracking technique allows a relatively faster increasing sequence  $\{\beta^{(k)}\}$  and improves the convergence.

## 6.2 Solving Non-separable Objective Problem

To show that M-ADMM can solve the problem with non-separable objective, we consider the Latent Low-Rank Representation (LatLRR) problem [25] for affine subspace clustering

$$\min_{\mathbf{Z}, \mathbf{L}} \|\mathbf{Z}\|_* + \|\mathbf{L}\|_* + \frac{\lambda}{2} \|\mathbf{XZ} + \mathbf{LX} - \mathbf{X}\|_F^2, \text{ s.t. } \mathbf{1}^\top \mathbf{Z} = \mathbf{1}^\top, \quad (48)$$

where  $\lambda > 0$  and the constraint is due to the affine subspace structure of data  $\mathbf{X}$  [8]. The objective of (48) is non-separable and

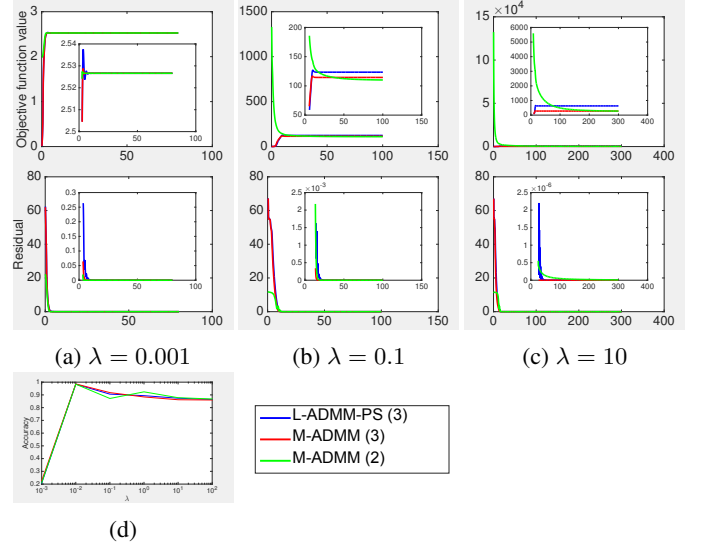


Fig. 4: Comparison of L-ADMM-PS (3), M-ADMM (3) and M-ADMM (2) on different choices of  $\lambda$ : (a)  $\lambda = 0.001$ ; (b)  $\lambda = 0.1$  and (c)  $\lambda = 10$ . Top row: plots of  $f(\mathbf{x}^k)$  v.s. CPU time; middle row: plots of  $\|\mathbf{Ax}^k - \mathbf{b}\|$  v.s. CPU time. (d) Subspace clustering accuracy v.s.  $\lambda$ . In (a)-(c), for better visualization, we plot the objective value and residual within a relatively smaller range of CPU time in subfigures.

can be rewritten as the following one with separable objective

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{L}, \mathbf{E}} \quad & \|\mathbf{Z}\|_* + \|\mathbf{L}\|_* + \frac{\lambda}{2} \|\mathbf{E}\|_F^2, \\ \text{s.t.} \quad & \mathbf{1}^\top \mathbf{Z} = \mathbf{1}^\top, \quad \mathbf{XZ} + \mathbf{LX} - \mathbf{X} = \mathbf{E}. \end{aligned} \quad (49)$$

We compare the following three solvers which own the convergence guarantee to solve the latent LRR problem:

- L-ADMM-PS (3): use (13) for 3 blocks problem (49).
- M-ADMM (3): use M-ADMM for 3 blocks problem (49).
- M-ADMM (2): use M-ADMM for 2 blocks problem (48).

Note that  $h(\mathbf{Z}, \mathbf{L}) = \frac{1}{2} \|\mathbf{XZ} + \mathbf{LX} - \mathbf{X}\|_F^2$  in (48) is  $\{2\|\mathbf{X}\|_2^2 \mathbf{I}, 2\|\mathbf{X}\|_2^2 \mathbf{I}\}$ -smooth. M-ADMM (2) uses the Lipschitz gradient surrogate in (22) to make the subproblems separable. For M-ADMM (3), we partition the three variables into two super blocks:  $\{\mathbf{Z}\}$  and  $\{\mathbf{L}, \mathbf{E}\}$ , and update them in the Gauss-Seidel way. In contrast, L-ADMM-PS updates  $\mathbf{Z}$ ,  $\mathbf{L}$  and  $\mathbf{E}$  in parallel.

We apply latent LRR for subspace clustering by using the learned  $\mathbf{Z}$  based on both the synthesized and real data. For the synthesized data, we generate  $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots]$  with its columns sampled from different subspaces. We construct  $k = 5$  independent subspaces  $\{\mathcal{S}_i\}_{i=1}^5 \subseteq \mathbb{R}^{200}$  whose bases  $\{\mathbf{U}_i\}_{i=1}^5$  are computed by  $\mathbf{U}_i = \mathbf{T}\mathbf{U}_i$ ,  $1 \leq i \leq 4$ , where  $\mathbf{T}$  is a random rotation and  $\mathbf{U}_1 \in \mathbb{R}^{200 \times 5}$  is a random orthogonal matrix. We sample 100 vectors from each subspace by  $\mathbf{X}_i = \mathbf{U}_i \mathbf{Q} + 0.1$ ,  $1 \leq i \leq 5$  with  $\mathbf{Q} \in \mathbb{R}^{5 \times 100}$  being an i.i.d.  $N(0, 1)$  matrix. Furthermore, 20% of data vectors are chosen to be corrupted, e.g., for a data vector  $\mathbf{x}$  chosen to be corrupted, its observed vector is computed by adding Gaussian noise with zero mean and variance  $0.2\|\mathbf{x}\|$ . Given  $\mathbf{X} \in \mathbb{R}^{200 \times 500}$  by the above way, we can solve the latent LRR problem by the three solvers and obtain the solution  $\mathbf{Z}^*$ . Then the data vectors can be grouped into  $k$  groups based on the affinity matrix  $(|\mathbf{Z}^*| + |\mathbf{Z}^*|^\top)/2$  by spectral clustering [25]. The clustering accuracy is used to evaluate the clustering

TABLE 3: Comparison of L-ADMM-PS (3) and M-ADMM (3) and M-ADMM (2) for latent LRR on the Hopkins 155 dataset.

Methods	L-ADMM-PS (3)	M-ADMM (3)	M-ADMM (2)
Accuracy (%)	90.9	<b>92.7</b>	87.1
CPU Time (s)	756.2	<b>738.5</b>	932.1

performance [25]. We test on different choices of  $\lambda$  and compare the three solvers based on  $f(\mathbf{x}^k)$  v.s. CPU time (in seconds),  $\|\mathbf{Ax} - \mathbf{b}\|$  v.s. CPU time and clustering accuracy. The results are shown in Figure 4 and we have the following observations:

- M-ADMM (3) always outperforms L-ADMM-PS (3) in the sense that the objective value is smaller when the algorithms converge and the residual decreases much faster. Both solve the same problem (49) with 3 blocks of variables. But M-ADMM (3) updates  $\mathbf{Z}$  and  $\{\mathbf{L}, \mathbf{E}\}$  sequentially, and thus it is faster than L-ADMM-PS (3) which updates them in parallel. This is consistent with our analysis at the end of Section 5.1.
- When  $\lambda$  is relatively small, M-ADMM (2) converges faster than M-ADMM (3). When  $\lambda$  is relatively large, M-ADMM (2) leads to a smaller objective value, but it requires much more running time (many more iterations). Both solvers have their advantages and disadvantages. In this experiment, the block number  $n$  and the looseness of the surrogate are two crucial factors. M-ADMM (2) solves (48) with only 2 blocks, but it requires constructing the Lipschitz gradient surrogate by (22) for  $h(\mathbf{Z}, \mathbf{L}) = \frac{\lambda}{2} \|\mathbf{XZ} + \mathbf{LX} - \mathbf{X}\|_F^2$ . This surrogate is looser when  $\lambda$  is larger. This is why M-ADMM (2) is slower when  $\lambda$  increases (the same phenomenon also appears in ISTA and FISTA [1]). On the other hand, M-ADMM (3) for 3 blocks problem (49) converges quickly regardless of the choice of  $\lambda$ . The issue of M-ADMM (3) is that the surrogate  $\hat{r}_i^k(\mathbf{x}_i)$  in (37)-(38) also becomes looser when  $\beta^{(k)}$  increases. So M-ADMM (3) may quickly get stuck and the final objective value is larger than M-ADMM (2). In practice, one has to balance the effects of both the block number  $n$  and the looseness of the surrogate, by considering the specific problems.

We further apply latent LRR for motion segmentation and test on the Hopkins 155 dataset [37]. This dataset contains 156 sequences, each with 39~550 vectors drawn from two or three motions (one motion corresponds to one subspace). Each sequence is a sole segmentation (clustering) task and thus there are 156 clustering tasks in total. We follow the experimental settings in [25] but without the complex post-processing. We set  $\lambda = 500$  and compare the performance by using M-ADMM (2), L-ADMM-PS (3) and M-ADMM (3). We stop the algorithms when

$$\|\mathbf{Ax}^k - \mathbf{b}\|/\|\mathbf{b}\| \leq \epsilon, \text{ and } \|\mathbf{x}^{k+1} - \mathbf{x}^k\|/\|\mathbf{b}\| \leq \epsilon, \quad (50)$$

where  $\epsilon = 10^{-4}$ . For each motion sequence, we record the clustering accuracy and the CPU time of solvers. Then the mean clustering accuracy and the total CPU time of all 156 sequences are reported in Table 3. It can be seen that, due to the same stopping criteria in (50), the CPU time of L-ADMM-PS (3) and that of M-ADMM (3) are similar. But the solution to latent LRR obtained by M-ADMM (3) achieves better clustering accuracy than L-ADMM-PS (3). The reason is that M-ADMM (3) obtains



Fig. 5: Images used for nonnegative matrix completion.

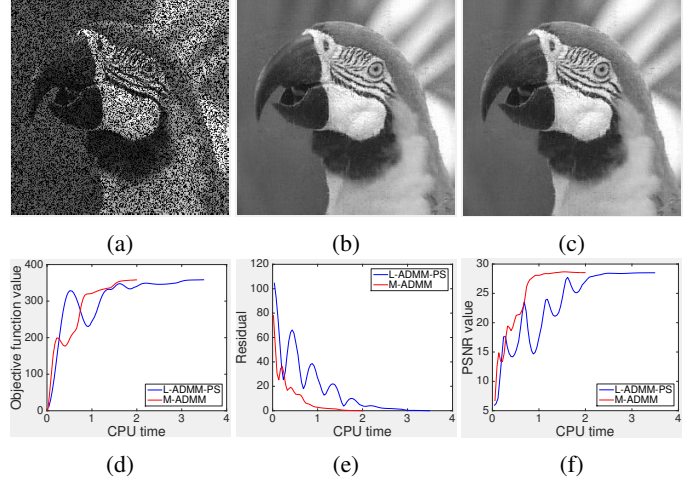


Fig. 6: Top row: the observed noisy image (left), recovered image by L-ADMM-PS (middle), and recovered image by M-ADMM (right). Bottom row: plots of  $f(\mathbf{x}^k)$  v.s. CPU time (left), plots of  $\|\mathbf{Ax}^k - \mathbf{b}\|$  v.s. CPU time (middle), and PSNR values v.s. CPU time (right).

a better solution with much smaller objective value within similar running time (or similar number of iterations). In this experiment, M-ADMM (2) for (48) is inferior to the other two solvers since the used  $\lambda$  is relatively large and thus the used majorant surrogate is loose.

### 6.3 Solving Nonnegative Matrix Completion

In this subsection, we show how to use Gauss-Seidel ADMM to solve a class of problems ( $n > 2$ ) with the condition (45) being satisfied. We consider the following nonnegative noisy matrix completion problem [27]

$$\min_{\mathbf{X}, \mathbf{E}} \|\mathbf{X}\|_* + \frac{\lambda}{2} \|\mathbf{E}\|^2, \text{ s.t. } \mathcal{P}_\Omega(\mathbf{X}) + \mathbf{E} = \mathbf{B}, \mathbf{X} \geq \mathbf{0}, \quad (51)$$

where  $\Omega$  is an index set and  $\mathcal{P}_\Omega$  is a linear mapping that keeps the entries in  $\Omega$  unchanged and those outside  $\Omega$  zeros. The above problem can be reformulated as a 3 blocks problem by (94) in [27] and then solved by L-ADMM-PS. We instead reformulate (51) as

$$\min_{\mathbf{X}, \mathbf{E}, \mathbf{Z}} \|\mathbf{X}\|_* + \frac{\lambda}{2} \|\mathbf{E}\|^2, \quad (52)$$

$$\text{s.t. } \mathcal{P}_\Omega(\mathbf{Z}) + \mathbf{E} = \mathbf{B}, \mathbf{X} = \mathbf{Z}, \mathbf{Z} \geq \mathbf{0}.$$

Note that (45) holds for (52) with the partition  $\{\mathbf{X}, \mathbf{E}\}$  and  $\{\mathbf{Z}\}$ . Thus (52) can be solved using (35)-(36) with closed form solutions for each variable. We still refer to this method as M-ADMM in this experiment.

We consider the same image inpainting problem in [27] which is to fill in the missing pixel values of a corrupted image. As the pixel values are nonnegative, the image inpainting problem can be solved by (51). The corrupted image is generated from the original image by sampling 60% of the pixels uniformly at random and adding Gaussian noise with mean zero and standard deviation 0.1. We use the same adaptive

TABLE 4: Numerical comparison on the image inpainting.

images	L-ADMM-PS			M-ADMM		
	PSNR	CPU	# Iter.	PSNR	CPU	# Iter.
parrot	28.51	3.50	87	28.54	2.00	55
barbara	27.69	3.36	85	27.72	2.27	60
boat	28.91	3.54	85	28.93	2.21	58
cameraman	26.06	3.33	84	26.08	2.15	58
foreman	31.83	3.80	86	31.84	2.06	54
house	31.26	3.48	87	31.26	2.29	56
lena	27.65	3.55	85	27.68	2.33	62
monarch	25.29	3.47	85	25.33	2.52	63

penalty to update  $\beta^{(k)}$  as [27]. We set  $\lambda = 10$ ,  $\epsilon_1 = 10^{-3}$ ,  $\epsilon_2 = 10^{-4}$  and  $\beta^{(0)} = \min(d_1, d_2)\epsilon_2$ , where  $d_1 \times d_2$  is the size of  $\mathbf{X}$ . We update  $\beta^{(k+1)} = \max(10\beta^{(k)}, 10^6)$  when  $\max_i(\beta^{(k)}\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|/\|\mathbf{b}\|) \leq \epsilon_1$ . The stopping criteria are  $\max_i(\|\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\|/\|\mathbf{b}\|) \leq \epsilon_2$  and  $\|\mathbf{Ax}^k - \mathbf{b}\|/\|\mathbf{b}\| \leq \epsilon_1$ . We test on 8 images, all with size  $256 \times 256$ , in Figure 5 and evaluate the recovery performance based on the PSNR value. The higher PSNR value indicates better recovery performance. The quantitative results are reported in Table 4 and Figure 6 gives more results test on the parrot image. It can be seen that, with slightly better recovery performance, M-ADMM converges faster than L-ADMM-PS. The improvement benefits from the sequential updating of  $\{\mathbf{X}\}$  and  $\{\mathbf{Z}, \mathbf{E}\}$  and avoids computing of the majorant surrogate as that in L-ADMM-PS.

## 7 CONCLUSIONS

This paper revisits ADMM, an old but reborn method for convex problems with linear constraint. Many previous ADMMs can be categorized into the Gauss-Seidel ADMMs and Jacobian ADMMs according to different updating orders of the primal variables. We observed that many previous ADMMs update the primal variables by minimizing different majorant functions. Then we proposed the majorant first-order surrogate functions and presented the unified frameworks with unified convergence analysis. They not only draw the connections with existing ADMMs, but also can be used to solve new problems with non-separable objectives. The convergence bound show that the convergence speed depends on the tightness of the used majorant functions. We then analyzed how to improve the tightness to improve the efficiency. We improve Jacobian ADMMs by introducing the Mixed Gauss-Seidel and Jacobian ADMM and the backtracking technique. We also discussed how to perform variable partition for efficient implementations. Experiments on both synthesized and real-world data well demonstrated the effectiveness of our new ADMMs.

In the future, one may consider extending our unified analysis based on MM to develop new ADMMs or solve other problems, e.g., strongly convex or nonconvex problems, and other ADMMs, e.g., stochastic ADMMs.

## REFERENCES

- [1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2009.
- [2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
- [3] C. Chen, B. He, Y. Ye, and X. Yuan. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *Optimization Online*, 2013.
- [4] Y. Chen, S. Sanghavi, and H. Xu. Improved graph clustering. *IEEE Transactions on Information Theory*, 60(10):6440–6455, 2014.
- [5] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. In *ICCV*, pages 2439–2446. IEEE, 2011.
- [6] W. Deng, M.-J. Lai, and W. Yin. On the  $o(1/k)$  convergence and parallelization of the alternating direction method of multipliers. *arXiv:1312.3040*, 2013.
- [7] E. Elhamifar and R. Vidal. Robust classification using structured sparse representation. In *CVPR*, pages 1873–1879, 2011.
- [8] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *TPAMI*, 35(11):2765–2781, Nov 2013.
- [9] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.
- [10] T. Goldstein, B. O’Donoghue, S. Setzer, and R. Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623, 2014.
- [11] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.
- [12] B. He and X. Yuan. On the  $O(1/n)$  convergence rate of the Douglas-Rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.
- [13] B. He and X. Yuan. Block-wise alternating direction method of multipliers for multipleblock convex programming and beyond, 2015.
- [14] M. R. Hestenes. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320, 1969.
- [15] M. Hong and Z.-Q. Luo. On the linear convergence of the alternating direction method of multipliers. *arXiv:1208.3922*, 2012.
- [16] B. Huang, C. Mu, D. Goldfarb, and J. Wright. Provable models for robust low-rank tensor recovery. *Pacific Journal of Optimization*, 2015.
- [17] H. Ji, C. Liu, Z. Shen, and Y. Xu. Robust video denoising using low rank matrix completion. In *CVPR*, pages 1791–1798, 2010.
- [18] K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of computational and graphical statistics*, 9(1):1–20, 2000.
- [19] X. Liang, X. Ren, Z. Zhang, and Y. Ma. Repairing sparse low-rank texture. In *ECCV*, pages 482–495. 2012.
- [20] T. Lin, S. Ma, and S. Zhang. On the sublinear convergence rate of multi-block ADMM. *arXiv preprint arXiv:1408.4265*, 2014.
- [21] T. Lin, S. Ma, and S. Zhang. Iteration complexity analysis of multi-block admm for a family of convex minimization without strong convexity. *arXiv preprint arXiv:1504.03087*, 2015.
- [22] Z. Lin, R. Liu, and H. Li. Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning. *Machine Learning*, 99(2):287–325, 2015.
- [23] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In *NIPS*, 2011.
- [24] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *TPAMI*, 2013.
- [25] G. Liu and S. Yan. Latent low-rank representation for subspace segmentation and feature extraction. In *ICCV*, 2011.
- [26] J. Liu, S. Ji, and J. Ye. SLEP: Sparse learning with efficient projections. <http://www.public.asu.edu/~jye02/Software/SLEP>, 2009.
- [27] R. Liu, Z. Lin, and Z. Su. Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning. In *ACML*, 2013.
- [28] C. Lu, J. Feng, Z. Lin, and S. Yan. Correlation adaptive subspace segmentation by trace Lasso. In *ICCV*, 2013.
- [29] C. Lu, H. Li, Z. Lin, and S. Yan. Fast proximal linearized alternating direction method of multiplier with parallel splitting. 2016.
- [30] C. Lu, S. Yan, and Z. Lin. Convex sparse spectral clustering: Single-view to multi-view. *TIP*, 25(6):2833–2843, 2016.
- [31] J. Mairal. Optimization with first-order surrogate functions. In *ICML*, 2013.
- [32] J. Mairal, F. Bach, J. Ponce, G. Sapiro, R. Jenatton, and G. Obozinski. SPAMS: Sparse modeling software. <http://spams-devel.gforge.inria.fr/index.html>, 2011.
- [33] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2004.
- [34] H. Ouyang, N. He, L. Tran, and A. Gray. Stochastic alternating direction method of multipliers. In *ICML*, 2013.
- [35] S. Oymak, A. Jalali, M. Fazel, Y. C. Eldar, and B. Hassibi. Simultaneously structured models with application to sparse and low-rank matrices. *TIT*, 61(5):2886–2908, 2015.
- [36] X. Shen and Y. Wu. A unified approach to salient object detection via low rank matrix recovery. In *CVPR*, pages 853–860, 2012.
- [37] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *CVPR*, 2007.
- [38] R. K. Vinayak, S. Oymak, and B. Hassibi. Graph clustering with missing data: Convex algorithms and analysis. In *NIPS*, pages 2996–3004, 2014.

- [39] X. Wang, M. Hong, S. Ma, and Z.-Q. Luo. Solving multiple-block separable convex minimization problems using two-block alternating direction method of multipliers. *arXiv:1308.5294*, 2013.
- [40] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *TPAMI*, 31(2):210–227, 2009.
- [41] R. Xia, Y. Pan, L. Du, and J. Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*, pages 2149–2155, 2014.
- [42] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, and N. Yu. Non-negative low rank and sparse graph for semi-supervised learning. In *CVPR*, pages 2328–2335. IEEE, 2012.

## APPENDIX

**Lemma 4.** Given any  $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$  and  $\mathbf{G} \succeq 0$  of compatible sizes, we have

$$\begin{aligned} & \langle \mathbf{a} - \mathbf{b}, \mathbf{c} - \mathbf{a} \rangle_{\mathbf{G}} \\ &= \frac{1}{2} (\|\mathbf{b} - \mathbf{c}\|_{\mathbf{G}}^2 - \|\mathbf{a} - \mathbf{c}\|_{\mathbf{G}}^2 - \|\mathbf{a} - \mathbf{b}\|_{\mathbf{G}}^2), \end{aligned} \quad (53)$$

$$\begin{aligned} & \langle \mathbf{a} - \mathbf{b}, \mathbf{c} - \mathbf{d} \rangle \\ &= \frac{1}{2} (\|\mathbf{a} - \mathbf{d}\|^2 - \|\mathbf{a} - \mathbf{c}\|^2 - \|\mathbf{b} - \mathbf{d}\|^2 + \|\mathbf{b} - \mathbf{c}\|^2). \end{aligned} \quad (54)$$

**Lemma 5. (Combination Rules for Majorant First-Order Surrogates)** Let  $\hat{f} \in \mathcal{S}_{\{\mathbf{L}_i\}_{i=1}^n}(f, \kappa)$  and  $\hat{f}' \in \mathcal{S}_{\{\mathbf{L}'_i\}_{i=1}^n}(f', \kappa)$ . Then the following combination rules hold:

- **Linear combination:** for any  $\alpha, \alpha' > 0$ ,  $\alpha f + \alpha' f'$  is a majorant surrogate function in  $\mathcal{S}_{\{\alpha \mathbf{L}_i + \alpha' \mathbf{L}'_i\}_{i=1}^n}(\alpha f + \alpha' f', \kappa)$ ;
- **Transitivity:** let  $F \in \mathcal{S}_{\{\mathbf{L}'_i\}_{i=1}^n}(\hat{f}, \kappa)$ . Then  $F$  is a majorant surrogate in  $\mathcal{S}_{\{\mathbf{L}_i + \mathbf{L}'_i\}_{i=1}^n}(f, \kappa)$ .

**Proof of Lemma 2.** We deduce

$$\begin{aligned} & f(\mathbf{x}) \stackrel{\textcircled{1}}{\leq} \hat{f}(\mathbf{x}) \\ &= \left( \hat{f}(\mathbf{x}) - \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \kappa_i\|_{\mathbf{P}_i}^2 \right) + \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \kappa_i\|_{\mathbf{P}_i}^2 \\ &\stackrel{\textcircled{2}}{\leq} \left( \hat{f}(\mathbf{y}) - \frac{1}{2} \sum_{i=1}^n \|\mathbf{y}_i - \kappa_i\|_{\mathbf{P}_i}^2 \right) - \langle \mathbf{u}, \mathbf{y} - \mathbf{x} \rangle \\ &\quad + \sum_{i=1}^n \langle \mathbf{x}_i - \kappa_i, \mathbf{y}_i - \mathbf{x}_i \rangle_{\mathbf{P}_i} + \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \kappa_i\|_{\mathbf{P}_i}^2 \\ &\stackrel{\textcircled{3}}{\leq} \left( f(\mathbf{y}) + \frac{1}{2} \sum_{i=1}^n \|\mathbf{y}_i - \kappa_i\|_{\mathbf{L}_i - \mathbf{P}_i}^2 \right) - \langle \mathbf{u}, \mathbf{y} - \mathbf{x} \rangle \\ &\quad + \sum_{i=1}^n \langle \mathbf{x}_i - \kappa_i, \mathbf{y}_i - \mathbf{x}_i \rangle_{\mathbf{P}_i} + \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \kappa_i\|_{\mathbf{P}_i}^2 \\ &\stackrel{\textcircled{4}}{=} \left( f(\mathbf{y}) + \frac{1}{2} \sum_{i=1}^n \|\mathbf{y}_i - \kappa_i\|_{\mathbf{L}_i - \mathbf{P}_i}^2 \right) - \langle \mathbf{u}, \mathbf{y} - \mathbf{x} \rangle \\ &\quad - \frac{1}{2} \sum_{i=1}^n (\|\mathbf{x}_i - \kappa_i\|_{\mathbf{P}_i}^2 + \|\mathbf{y}_i - \mathbf{x}_i\|_{\mathbf{P}_i}^2 - \|\mathbf{y}_i - \kappa_i\|_{\mathbf{P}_i}^2) \\ &\quad + \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \kappa_i\|_{\mathbf{P}_i}^2 \\ &= f(\mathbf{y}) - \langle \mathbf{u}, \mathbf{y} - \mathbf{x} \rangle + \frac{1}{2} \sum_{i=1}^n (\|\mathbf{y}_i - \kappa_i\|_{\mathbf{L}_i}^2 - \|\mathbf{y}_i - \mathbf{x}_i\|_{\mathbf{P}_i}^2), \end{aligned}$$

where ① is from the fact that  $\hat{f}$  is a majorant function of  $f$ , ② is from the convexity of  $\hat{f}(\mathbf{x}) - \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \kappa_i\|_{\mathbf{P}_i}^2$  (or  $\hat{f}$  is  $\{\mathbf{P}_i\}_{i=1}^n$ -strongly convex), ③ uses (18), and ④ is from (53). ■

**Proof of Lemma 3.** By using (53), for any  $\mathbf{x}$  and  $\mathbf{y}$ , we have

$$\frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|^2 - \frac{1}{2} \|\mathbf{Ay} - \mathbf{b}\|^2 \quad (55)$$

$$= \frac{1}{2} \|\mathbf{A}(\mathbf{x} - \mathbf{y})\|^2 + \langle \mathbf{A}(\mathbf{x} - \mathbf{y}), \mathbf{Ay} - \mathbf{b} \rangle$$

$$\leq \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{L}'_i}^2 + \langle \mathbf{A}(\mathbf{x} - \mathbf{y}), \mathbf{Ay} - \mathbf{b} \rangle \quad (56)$$

$$\leq \frac{1}{2} \sum_{i=1}^n (\|\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{G}_i}^2 + \|\mathbf{A}_i(\mathbf{x}_i - \mathbf{y}_i)\|^2) \quad (57)$$

$$\begin{aligned} & + \sum_{i=1}^n \langle \mathbf{A}_i(\mathbf{x}_i - \mathbf{y}_i), \mathbf{Ay} - \mathbf{b} \rangle \\ &= \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{G}_i}^2 \\ & + \frac{1}{2} \sum_{i=1}^n (\|\mathbf{A}_i(\mathbf{x}_i - \mathbf{y}_i) + \mathbf{Ay} - \mathbf{b}\|^2 - \|\mathbf{Ay} - \mathbf{b}\|^2) \\ &= \frac{1}{2} \sum_{i=1}^n \left( \|\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{G}_i}^2 + \left\| \mathbf{A}_i \mathbf{x}_i + \sum_{j \neq i} \mathbf{A}_j \mathbf{y}_j - \mathbf{b} \right\|^2 \right) \\ & - \frac{n}{2} \|\mathbf{Ay} - \mathbf{b}\|^2, \end{aligned}$$

where (56) holds for some  $\mathbf{L}'_i$ 's; e.g., we can choose  $\mathbf{L}'_i \succeq n \mathbf{A}_i^\top \mathbf{A}_i$ , and (57) uses  $\mathbf{G}_i \succeq \mathbf{L}'_i - \mathbf{A}_i^\top \mathbf{A}_i$ . Note that  $r(\mathbf{x})$  is convex and (55)-(56) imply that (17) holds. Thus,  $r$  is  $\{\mathbf{L}'_i\}_{i=1}^n$ -smooth. By the definition of  $\hat{r}$  in (21), the above inequality implies that  $r(\mathbf{x}) \leq \hat{r}(\mathbf{x})$ . Furthermore, it is easy to obtain (22) by substituting  $\mathbf{G}_i = \eta_i \mathbf{I} - \mathbf{A}_i^\top \mathbf{A}_i$  into (57). ■

We give the proof of Theorem 4. In the following, we define

$$\hat{\lambda}^{k+1} = \lambda^k + \beta^{(k)} (\mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^k - \mathbf{b}). \quad (58)$$

**Proposition 1.** In Algorithm 4, under the assumptions of Theorem 4, for any  $\mathbf{x}$ , we have

$$\begin{aligned} & f(\mathbf{x}^{k+1}) - f(\mathbf{x}) - \langle \mathbf{A}^\top \hat{\lambda}^{k+1}, \mathbf{x} - \mathbf{x}^{k+1} \rangle \\ &\leq \frac{\beta^{(k)}}{2} \sum_{j=1}^2 \left( \|\mathbf{x}_{B_j} - \mathbf{x}_{B_j}^k\|_{\mathbf{H}_j^k}^2 - \|\mathbf{x}_{B_j} - \mathbf{x}_{B_j}^{k+1}\|_{\mathbf{H}_j^{k+1}}^2 \right) \\ &\quad - \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2^k}^2, \end{aligned} \quad (59)$$

where  $\mathbf{H}_1^k = \text{Diag} \left\{ \frac{1}{\beta^{(k)}} \mathbf{L}_i + \mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^k, i \in B_1 \right\} - \mathbf{A}_{B_1}^\top \mathbf{A}_{B_1}$ ,  $\mathbf{H}_2^k = \text{Diag} \left\{ \frac{1}{\beta^{(k)}} \mathbf{L}_i + \mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^k, i \in B_2 \right\}$  and  $\mathbf{K}_2^k = \text{Diag} \{ \mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^k, i \in B_2 \}$ .

**Proposition 2.** In Algorithm 4, for any  $\lambda$ , we have

$$\begin{aligned} & \langle \mathbf{Ax}^{k+1} - \mathbf{b}, \lambda - \hat{\lambda}^{k+1} \rangle + \frac{\beta^{(0)} \alpha}{2} \|\mathbf{Ax}^{k+1} - \mathbf{b}\|^2 \\ &\leq \frac{\beta^{(k)}}{2} \left( \|\lambda - \lambda^k\|_{\mathbf{H}_3^k}^2 - \|\lambda - \lambda^{k+1}\|_{\mathbf{H}_3^{k+1}}^2 \right) \\ &\quad + \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2^k}^2, \end{aligned} \quad (60)$$

where  $\mathbf{H}_3^k = \left( 1/\beta^{(k)} \right)^2 \mathbf{I}$  and  $\alpha = \min \left\{ \frac{1}{2}, \frac{\tau}{2 \|\mathbf{A}_{B_2}\|_2^2} \right\}$ .



**Proof of Theorem 4.** Let  $\mathbf{x} = \mathbf{x}^*$  and  $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$  in (59) and (60). We have

$$\begin{aligned}
& f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*) + \langle \mathbf{A}^\top \boldsymbol{\lambda}^*, \mathbf{x}^{k+1} - \mathbf{x}^* \rangle \\
& + \frac{\beta^{(0)}\alpha}{2} \|\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}\|^2 \\
& \leq \langle \mathbf{A}^\top (\boldsymbol{\lambda}^* - \hat{\boldsymbol{\lambda}}^{k+1}), \mathbf{x}^{k+1} - \mathbf{x}^* \rangle - \langle \mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}, \boldsymbol{\lambda}^* - \hat{\boldsymbol{\lambda}}^{k+1} \rangle \\
& + \frac{\beta^{(k)}}{2} \left( \sum_{i=j}^2 \|\mathbf{x}_{B_j}^* - \mathbf{x}_{B_j}^k\|_{\mathbf{H}_j^k}^2 + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k\|_{\mathbf{H}_3^k}^2 \right) \\
& - \frac{\beta^{(k)}}{2} \left( \sum_{i=j}^2 \|\mathbf{x}_{B_j}^* - \mathbf{x}_{B_j}^{k+1}\|_{\mathbf{H}_j^{k+1}}^2 + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^{k+1}\|_{\mathbf{H}_3^{k+1}}^2 \right) \\
& = \frac{\beta^{(k)}}{2} \left( \sum_{i=j}^2 \|\mathbf{x}_{B_j}^* - \mathbf{x}_{B_j}^k\|_{\mathbf{H}_j^k}^2 + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k\|_{\mathbf{H}_3^k}^2 \right) \\
& - \frac{\beta^{(k)}}{2} \left( \sum_{i=j}^2 \|\mathbf{x}_{B_j}^* - \mathbf{x}_{B_j}^{k+1}\|_{\mathbf{H}_j^{k+1}}^2 + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^{k+1}\|_{\mathbf{H}_3^{k+1}}^2 \right),
\end{aligned}$$

where the last equation uses the fact  $\mathbf{A}\mathbf{x}^* = \mathbf{b}$ . Note that  $\sum_{k=0}^K \gamma^{(k)} = 1$ . Multiplying  $\gamma^{(k)}$  on both sides of the above inequalities and summing them from 0 to  $K$ , we have

$$\begin{aligned}
& \sum_{k=0}^K \gamma^{(k)} f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*) + \left\langle \mathbf{A}^\top \boldsymbol{\lambda}^*, \sum_{k=0}^K \gamma^{(k)} \mathbf{x}^{k+1} - \mathbf{x}^* \right\rangle \\
& + \frac{\beta^{(0)}\alpha}{2} \sum_{k=0}^K \gamma^{(k)} \|\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}\|^2 \\
& \leq \frac{\sum_{j=1}^2 \|\mathbf{x}_{B_j}^* - \mathbf{x}_{B_j}^0\|_{\mathbf{H}_j^0}^2 + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\|_{\mathbf{H}_3^0}^2}{2 \sum_{k=0}^K (\beta^{(k)})^{-1}}.
\end{aligned}$$

By the definition of  $\bar{\mathbf{x}}^K$  and the convexity of  $f$  and  $\|\cdot\|^2$ , we have

$$\begin{aligned}
& f(\bar{\mathbf{x}}^K) - f(\mathbf{x}^*) + \langle \mathbf{A}^\top \boldsymbol{\lambda}^*, \bar{\mathbf{x}}^K - \mathbf{x}^* \rangle + \frac{\beta^{(0)}\alpha}{2} \|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\|^2 \\
& \leq \sum_{k=0}^K \gamma^{(k)} f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*) + \left\langle \mathbf{A}^\top \boldsymbol{\lambda}^*, \sum_{k=0}^K \gamma^{(k)} \mathbf{x}^{k+1} - \mathbf{x}^* \right\rangle \\
& + \frac{\beta^{(0)}\alpha}{2} \sum_{k=0}^K \gamma^{(k)} \|\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}\|^2 \\
& \leq \frac{\sum_{j=1}^2 \|\mathbf{x}_{B_j}^* - \mathbf{x}_{B_j}^0\|_{\mathbf{H}_j^0}^2 + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\|_{\mathbf{H}_3^0}^2}{2 \sum_{k=0}^K (\beta^{(k)})^{-1}}.
\end{aligned}$$

The proof is completed.  $\blacksquare$

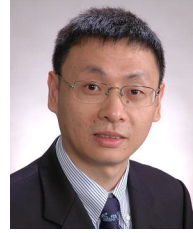


**Canyi Lu** received the bachelor degree in mathematics from the Fuzhou University in 2009, and the master degree in the pattern recognition and intelligent system from the University of Science and Technology of China in 2012. He is currently a Ph.D. student with the Department of Electrical and Computer Engineering at the National University of Singapore. His current research interests include computer vision, machine learning, pattern recognition and optimization. He was the winner of the Microsoft Research Asia Fellowship

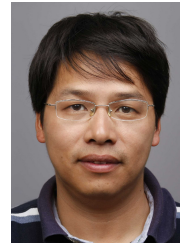
ship 2014.



**Jiashi Feng** is currently an assistant Professor in the department of electrical and computer engineering in the National University of Singapore. He got his B.E. degree from University of Science and Technology, China in 2007 and Ph.D. degree from National University of Singapore in 2014. He was a postdoc researcher at University of California from 2014 to 2015. His current research interest focus on machine learning and computer vision techniques for large-scale data analysis. Specifically, he has done work in object recognition, deep learning, machine learning, high-dimensional statistics and big data analysis.



**Shuicheng Yan** is currently an Associate Professor at the Department of Electrical and Computer Engineering at National University of Singapore, and the founding lead of the Learning and Vision Research Group (<http://www.lv-nus.org>). Dr. Yan's research areas include machine learning, computer vision and multimedia, and he has authored/co-authored hundreds of technical papers over a wide range of research topics, with Google Scholar citation >30,000 times and H-index 64. He is ISI Highly-cited Researcher, 2014 and IAPR Fellow 2014. He has been serving as an associate editor of IEEE TKDE, TCSVT and ACM Transactions on Intelligent Systems and Technology (ACM TIST). He received the Best Paper Awards from ACM MM'13 (Best Paper and Best Student Paper), ACM MM12 (Best Demo), PCM'11, ACM MM10, ICME10 and ICIMCS'09, the runner-up prize of ILSVRC'13, the winner prize of ILSVRC14 detection task, the winner prizes of the classification task in PASCAL VOC 2010-2012, the winner prize of the segmentation task in PASCAL VOC 2012, the honourable mention prize of the detection task in PASCAL VOC'10, 2010 TCSVT Best Associate Editor (BAE) Award, 2010 Young Faculty Research Award, 2011 Singapore Young Scientist Award, and 2012 NUS Young Researcher Award.



**Zhouchen Lin** received the Ph.D. degree in Applied Mathematics from Peking University, in 2000. He is currently a Professor at Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University. He is also a Chair Professor at Northeast Normal University and a Guest Professor at Beijing Jiaotong University. Before March 2012, he was a Lead Researcher at Visual Computing Group, Microsoft Research Asia. He was a Guest Professor at Shanghai Jiaotong University and Southeast University, and a Guest Researcher at Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer vision, image processing, computer graphics, machine learning, pattern recognition, and numerical computation and optimization. He is an Associate Editor of IEEE Trans. Pattern Analysis and Machine Intelligence and International J. Computer Vision, an area chair of CVPR 2014, ICCV 2015, NIPS 2015 and AAAI 2016, and a Senior Member of the IEEE.

# Supplementary Material

This document contains two parts. First, we give the proofs of some lemmas and propositions which are used to prove Theorem 4. Second, we give the implementation details of some problems in the experiments.

## 1. Proofs

**Proof of Lemma 5.** Lemma 5 is obvious by using the definition of the majorant first order surrogate function and the following lemma.

**Lemma 6.** Let  $f, f' : \mathbb{R}^{p_1} \times \cdots \times \mathbb{R}^{p_n} \rightarrow \mathbb{R}$  be convex, and  $\{\mathbf{L}_i\}_{i=1}^n$ -smooth and  $\{\mathbf{L}'_i\}_{i=1}^n$ -smooth, respectively. We only consider two cases: (1) if  $\mathbf{L}_i \succeq \mathbf{L}'_i$ , define  $\max\{\mathbf{L}_i, \mathbf{L}'_i\} = \mathbf{L}_i$ ; (2) if  $\mathbf{L}'_i \succeq \mathbf{L}_i$ , define  $\max\{\mathbf{L}_i, \mathbf{L}'_i\} = \mathbf{L}'_i$ . Then  $f - f'$  is  $\{\max\{\mathbf{L}_i, \mathbf{L}'_i\}\}_{i=1}^n$ -smooth, and  $f + f'$  is  $\{\mathbf{L}_i + \mathbf{L}'_i\}_{i=1}^n$ -smooth.

**Proof of Lemma 6.** Let  $h = f - f'$ . By using (17) and the convexity of  $f$  and  $f'$ , for any  $\mathbf{x} = [\mathbf{x}_1; \cdots; \mathbf{x}_n]$  and  $\mathbf{y} = [\mathbf{y}_1; \cdots; \mathbf{y}_n]$  with  $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^{p_i}$ ,  $i = 1, \cdots, n$ , we have

$$\begin{aligned} 0 &\leq f(\mathbf{x}) - f(\mathbf{y}) - \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \leq \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{L}_i}^2, \\ -\frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{L}'_i}^2 &\leq -f'(\mathbf{x}) + f'(\mathbf{y}) + \langle \nabla f'(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \leq 0. \end{aligned}$$

Summing the above two inequalities we have

$$|h(\mathbf{x}) - h(\mathbf{y}) - \langle \nabla h(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle| \leq \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{y}_i\|_{\max\{\mathbf{L}_i, \mathbf{L}'_i\}}^2.$$

Thus  $h$  is  $\{\max\{\mathbf{L}_i, \mathbf{L}'_i\}\}_{i=1}^n$ -smooth. It is easy to see that  $f + f'$  is  $\{\mathbf{L}_i + \mathbf{L}'_i\}_{i=1}^n$ -smooth by applying (17) for  $f$  and  $f'$ .  $\blacksquare$

**Proof of Proposition 1.** First, for  $i \in B_1$ , by the optimality of  $\mathbf{x}_i^{k+1}$  to problem (39) in Algorithm 4, there exists  $\mathbf{u}_i^{k+1} \in \partial \hat{f}_i^k(\mathbf{x}_i^{k+1})$  such that

$$\begin{aligned} -\mathbf{u}_i^{k+1} &= \nabla \hat{r}_i^k(\mathbf{x}_i^{k+1}) \\ &\stackrel{\textcircled{1}}{=} \mathbf{A}_i^\top \left( \beta^{(k)} \mathbf{A}_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) + \beta^{(k)} (\mathbf{A} \mathbf{x}^k - \mathbf{b}) + \boldsymbol{\lambda}^k \right) + \beta^{(k)} \mathbf{G}_i^k(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) \\ &= \mathbf{A}_i^\top \left( \beta^{(k)} (\mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^k - \mathbf{b}) + \boldsymbol{\lambda}^k \right) - \beta^{(k)} \mathbf{A}_i^\top \mathbf{A}_{B_1}(\mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k) + \beta^{(k)} (\mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^k)(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) \\ &\stackrel{\textcircled{2}}{=} \mathbf{A}_i^\top \hat{\boldsymbol{\lambda}}^{k+1} - \beta^{(k)} \mathbf{A}_i^\top \mathbf{A}_{B_1}(\mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k) + \beta^{(k)} (\mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^k)(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k), \end{aligned}$$

where  $\textcircled{1}$  uses the definition of  $\hat{r}_i^k$  in (37), and  $\textcircled{2}$  uses the definition of  $\hat{\boldsymbol{\lambda}}^{k+1}$  in (58). A dot-product with  $\mathbf{x}_i^{k+1} - \mathbf{x}_i^k$  on both sides of the above equation gives

$$\begin{aligned} -\langle \mathbf{u}_{B_1}^{k+1}, \mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k \rangle &= -\sum_{i \in B_1} \langle \mathbf{u}_i^{k+1}, \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \rangle \\ &= \sum_{i \in B_1} \left\langle \mathbf{A}_i^\top \hat{\boldsymbol{\lambda}}^{k+1} - \beta^{(k)} \mathbf{A}_i^\top \mathbf{A}_{B_1}(\mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k), \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \right\rangle + \sum_{i \in B_1} \left\langle \beta^{(k)} (\mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^k)(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k), \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \right\rangle \\ &= \langle \mathbf{A}_{B_1}^\top \hat{\boldsymbol{\lambda}}^{k+1}, \mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k \rangle + \beta^{(k)} \langle \mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k, \mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k \rangle_{\mathbf{K}_1^k - \mathbf{A}_{B_1}^\top \mathbf{A}_{B_1}} \\ &= \langle \mathbf{A}_{B_1}^\top \hat{\boldsymbol{\lambda}}^{k+1}, \mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k \rangle + \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k\|_{\mathbf{K}_1^k - \mathbf{A}_{B_1}^\top \mathbf{A}_{B_1}}^2 + \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_1} - \mathbf{x}_{B_1}^{k+1}\|_{\mathbf{K}_1^k - \mathbf{A}_{B_1}^\top \mathbf{A}_{B_1}}^2 - \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_1} - \mathbf{x}_{B_1}^k\|_{\mathbf{K}_1^k - \mathbf{A}_{B_1}^\top \mathbf{A}_{B_1}}^2 \\ &\stackrel{\textcircled{1}}{\geq} \langle \mathbf{A}_{B_1}^\top \hat{\boldsymbol{\lambda}}^{k+1}, \mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k \rangle + \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_1} - \mathbf{x}_{B_1}^k\|_{\mathbf{K}_1^k - \mathbf{A}_{B_1}^\top \mathbf{A}_{B_1}}^2 - \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_1} - \mathbf{x}_{B_1}^{k+1}\|_{\mathbf{K}_1^k - \mathbf{A}_{B_1}^\top \mathbf{A}_{B_1}}^2, \end{aligned} \quad (61)$$

where  $\mathbf{K}_1^k = \text{Diag}\{\mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^k, i \in B_1\}$  and  $\textcircled{1}$  uses  $\|\mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1}^k\|_{\mathbf{K}_1^k - \mathbf{A}_{B_1}^\top \mathbf{A}_{B_1}}^2 \geq 0$  due to (42).

Second, for  $i \in B_2$ , by the optimality of  $\mathbf{x}_i^{k+1}$  to problem (40) in Algorithm 4, there exists  $\mathbf{u}_i^{k+1} \in \partial \hat{f}_i^k(\mathbf{x}_i^{k+1})$  such that

$$\begin{aligned} -\mathbf{u}_i^{k+1} &= \nabla \hat{r}_i^k(\mathbf{x}_i^{k+1}) \\ &= \mathbf{A}_i^\top \left( \beta^{(k)} \mathbf{A}_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) + \beta^{(k)} (\mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^k - \mathbf{b}) \right) \\ &\quad + \mathbf{A}_i^\top \boldsymbol{\lambda}^k + \beta^{(k)} \mathbf{G}_i^k(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) \\ &= \mathbf{A}_i^\top \hat{\boldsymbol{\lambda}}^{k+1} + \beta^{(k)} (\mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^k)(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k), \end{aligned}$$



where we use the definitions of  $\hat{\mathbf{r}}_i^k$  in (38) and  $\hat{\boldsymbol{\lambda}}^{k+1}$  in (58). A dot-product with  $\mathbf{x}_i^{k+1} - \mathbf{x}_i$  on both sides of the above equation gives

$$\begin{aligned}
& -\langle \mathbf{u}_{B_2}^{k+1}, \mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2} \rangle = -\sum_{i \in B_2} \langle \mathbf{u}_i^{k+1}, \mathbf{x}_i^{k+1} - \mathbf{x}_i \rangle \\
& = \sum_{i \in B_2} \left\langle \mathbf{A}_i^\top \hat{\boldsymbol{\lambda}}^{k+1} + \beta^{(k)} (\mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^k) (\mathbf{x}_i^{k+1} - \mathbf{x}_i^k), \mathbf{x}_i^{k+1} - \mathbf{x}_i \right\rangle \\
& = \langle \mathbf{A}_{B_2}^\top \hat{\boldsymbol{\lambda}}^{k+1}, \mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2} \rangle + \beta^{(k)} \langle \mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k, \mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1} \rangle_{\mathbf{K}_2^k} \\
& = \langle \mathbf{A}_{B_2}^\top \hat{\boldsymbol{\lambda}}^{k+1}, \mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2} \rangle - \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_2} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2^k}^2 + \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_2} - \mathbf{x}_{B_2}^{k+1}\|_{\mathbf{K}_2^k}^2 + \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2^k}^2, \tag{62}
\end{aligned}$$

where  $\mathbf{K}_2^k = \text{Diag}\{\mathbf{A}_i^\top \mathbf{A}_i + \mathbf{G}_i^k, i \in B_2\}$ .

Third, note that  $\hat{f}^k \in \mathcal{S}_{\{\mathbf{L}_i, \mathbf{P}_i\}_{i=1}^n}(f, \mathbf{x}^k)$ . By using (20), we have

$$\begin{aligned}
& f(\mathbf{x}^{k+1}) - f(\mathbf{x}) \\
& \leq \langle \mathbf{u}^{k+1}, \mathbf{x}^{k+1} - \mathbf{x} \rangle + \frac{1}{2} \sum_{i=1}^n \left( \|\mathbf{x}_i - \mathbf{x}_i^k\|_{\mathbf{L}_i}^2 - \|\mathbf{x}_i - \mathbf{x}_i^{k+1}\|_{\mathbf{P}_i}^2 \right) \\
& \leq \langle \mathbf{u}^{k+1}, \mathbf{x}^{k+1} - \mathbf{x} \rangle + \frac{1}{2} \sum_{i=1}^n \left( \|\mathbf{x}_i - \mathbf{x}_i^k\|_{\mathbf{L}_i}^2 - \|\mathbf{x}_i - \mathbf{x}_i^{k+1}\|_{\mathbf{L}_i}^2 \right) \\
& = \langle \mathbf{u}_{B_1}^{k+1}, \mathbf{x}_{B_1}^{k+1} - \mathbf{x}_{B_1} \rangle + \langle \mathbf{u}_{B_2}^{k+1}, \mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2} \rangle + \frac{1}{2} \sum_{j=1}^2 \left( \|\mathbf{x}_{B_j} - \mathbf{x}_{B_j}^k\|_{\mathbf{L}_{B_j}}^2 - \|\mathbf{x}_{B_j} - \mathbf{x}_{B_j}^{k+1}\|_{\mathbf{L}_{B_j}}^2 \right) \\
& \stackrel{\textcircled{1}}{\leq} -\langle \mathbf{A}^\top \hat{\boldsymbol{\lambda}}^{k+1}, \mathbf{x}^{k+1} - \mathbf{x} \rangle - \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2}^2 + \frac{\beta^{(k)}}{2} \sum_{j=1}^2 \left( \|\mathbf{x}_{B_j} - \mathbf{x}_{B_j}^k\|_{\mathbf{H}_j^k}^2 - \|\mathbf{x}_{B_j} - \mathbf{x}_{B_j}^{k+1}\|_{\mathbf{H}_j^{k+1}}^2 \right)
\end{aligned}$$

where  $\mathbf{L}_{B_j} = \text{Diag}\{\mathbf{L}_i, i \in B_j\}$  and  $\textcircled{1}$  uses (61)-(62), the definitions of  $\mathbf{H}_j^k$  in Proposition 1 and the fact  $\beta^{(k+1)} \geq \beta^{(k)}$ . The proof is completed.  $\blacksquare$

**Proof of Proposition 2.** By using line 4 of Algorithm 3, (54) and the fact that  $\beta^{(k+1)} \geq \beta^{(k)}$ , we have

$$\begin{aligned}
& \langle \mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}, \boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}}^{k+1} \rangle = \frac{1}{\beta^{(k)}} \langle \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k, \boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}}^{k+1} \rangle \\
& = \frac{1}{2\beta^{(k)}} \left( \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^k\|^2 - \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^{k+1}\|^2 \right) - \frac{1}{2\beta^{(k)}} \left( \|\hat{\boldsymbol{\lambda}}^{k+1} - \boldsymbol{\lambda}^k\|^2 - \|\boldsymbol{\lambda}^{k+1} - \hat{\boldsymbol{\lambda}}^{k+1}\|^2 \right) \\
& \quad - \frac{1}{2\beta^{(k)}} \left( \|\hat{\boldsymbol{\lambda}}^{k+1} - \boldsymbol{\lambda}^k\|^2 - \|\boldsymbol{\lambda}^{k+1} - \hat{\boldsymbol{\lambda}}^{k+1}\|^2 \right). \tag{63}
\end{aligned}$$

Consider the last two terms in (63). We deduce

$$\begin{aligned}
& \frac{1}{2\beta^{(k)}} \left( \|\hat{\boldsymbol{\lambda}}^{k+1} - \boldsymbol{\lambda}^k\|^2 - \|\boldsymbol{\lambda}^{k+1} - \hat{\boldsymbol{\lambda}}^{k+1}\|^2 \right) \\
& \stackrel{\textcircled{1}}{=} \frac{\beta^{(k)}}{2} \|\mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^k - \mathbf{b}\|^2 - \|\mathbf{A}_{B_2} (\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k)\|^2 \\
& = \frac{\beta^{(k)}}{2} \left( \|\mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^k - \mathbf{b}\|^2 - \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2^k}^2 \right) + \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2^k - \mathbf{A}_{B_2}^\top \mathbf{A}_{B_2}}^2 \\
& \stackrel{\textcircled{2}}{\geq} \frac{\beta^{(k)}}{2} \left( \|\mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^k - \mathbf{b}\|^2 - \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2^k}^2 \right) + \frac{\tau}{\|\mathbf{A}_{B_2}\|_2^2} \|\mathbf{A}_{B_2} (\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k)\|^2 \\
& \stackrel{\textcircled{3}}{\geq} \beta^{(k)} \alpha \|\mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^k - \mathbf{b}\|^2 - \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2^k}^2 + \beta^{(k)} \alpha \|\mathbf{A}_{B_2} (\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k)\|^2 \\
& \geq \frac{\beta^{(k)} \alpha}{2} \|\mathbf{A}_{B_1} \mathbf{x}_{B_1}^{k+1} + \mathbf{A}_{B_2} \mathbf{x}_{B_2}^{k+1} - \mathbf{b}\|^2 - \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2^k}^2 \\
& \stackrel{\textcircled{4}}{\geq} \frac{\beta^{(0)} \alpha}{2} \|\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}\|^2 - \frac{\beta^{(k)}}{2} \|\mathbf{x}_{B_2}^{k+1} - \mathbf{x}_{B_2}^k\|_{\mathbf{K}_2^k}^2, \tag{64}
\end{aligned}$$

where  $\textcircled{1}$  uses (4) and (58),  $\textcircled{2}$  uses (44),  $\textcircled{3}$  uses  $\alpha = \min\left\{\frac{1}{2}, \frac{\tau}{2\|\mathbf{A}_{B_2}\|_2^2}\right\}$ , and  $\textcircled{4}$  uses  $\beta^{(k)} \geq \beta^{(k-1)} \geq \dots \geq \beta^{(0)}$ . The proof is completed by substituting (64) into (63).  $\blacksquare$

## 2. Implementation Details

### 2.1 Latent Low-Rank Representation

Consider the following Latent Low-Rank Representation (LRR) problem

$$\min_{\mathbf{Z}, \mathbf{L}} \|\mathbf{Z}\|_* + \|\mathbf{L}\|_* + \frac{\lambda}{2} \|\mathbf{XZ} + \mathbf{LX} - \mathbf{X}\|_F^2, \text{ s.t. } \mathbf{1}^T \mathbf{Z} = \mathbf{1}^T, \quad (65)$$

Problem (65) is equivalent to

$$\min_{\mathbf{Z}, \mathbf{L}, \mathbf{E}} \|\mathbf{Z}\|_* + \|\mathbf{L}\|_* + \frac{\lambda}{2} \|\mathbf{E}\|_F^2, \text{ s.t. } \mathbf{1}^T \mathbf{Z} = \mathbf{1}^T, \mathbf{XZ} + \mathbf{LX} - \mathbf{X} = \mathbf{E}. \quad (66)$$

(a) Solve (65) by M-ADMM (2)

The augmented Lagrangian function of (65) is

$$\mathcal{L}(\mathbf{Z}, \mathbf{L}, \boldsymbol{\lambda}, \beta) = \|\mathbf{Z}\|_* + \|\mathbf{L}\|_* + \frac{\lambda}{2} \|\mathbf{XZ} + \mathbf{LX} - \mathbf{X}\|_F^2 + \langle \boldsymbol{\lambda}, \mathbf{1}^T \mathbf{Z} - \mathbf{1}^T \rangle + \frac{\beta}{2} \|\mathbf{1}^T \mathbf{Z} - \mathbf{1}^T\|^2.$$

It is easy to verify that  $\frac{1}{2} \|\mathbf{XZ} + \mathbf{LX} - \mathbf{X}\|_F^2$  is  $\{L_1 \mathbf{I}, L_2 \mathbf{I}\}$ -smooth, where  $L_1 = L_2 = 2\|\mathbf{X}\|_2^2$ , and  $\frac{1}{2} \|\mathbf{1}^T \mathbf{Z} - \mathbf{1}^T\|^2$  is  $\eta$ -smooth, where  $\eta > \|\mathbf{1}\|^2$ . By using these properties, M-ADMM (2) solves (65) by the following updating rules

$$\begin{cases} \mathbf{Z}^{k+1} = \arg \min_{\mathbf{Z}} \|\mathbf{Z}\|_* + \frac{\lambda L_1 + \beta^{(k)} \eta}{2} \left\| \mathbf{Z} - \mathbf{Z}^k + \frac{\lambda \mathbf{X}^T (\mathbf{XZ}^k + \mathbf{L}^k \mathbf{X} - \mathbf{X}) + \mathbf{1} (\beta^{(k)} (\mathbf{1}^T \mathbf{Z}^k - \mathbf{1}^T) + \boldsymbol{\lambda}^k)}{\lambda L_1 + \beta^{(k)} \eta} \right\|_F^2, \\ \mathbf{L}^{k+1} = \arg \min_{\mathbf{L}} \|\mathbf{L}\|_* + \frac{\lambda L_2}{2} \left\| \mathbf{L} - \mathbf{L}^k + \frac{(\mathbf{XZ}^k + \mathbf{L}^k \mathbf{X} - \mathbf{X}) \mathbf{X}^T}{L_2} \right\|_F^2, \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta^{(k)} (\mathbf{1}^T \mathbf{Z}^{k+1} - \mathbf{1}^T). \end{cases}$$

(b) Solve (66) by L-ADMM-PS (3)

The augmented Lagrangian function of (66) is

$$\begin{aligned} \mathcal{L}(\mathbf{Z}, \mathbf{L}, \boldsymbol{\lambda}, \beta) = & \|\mathbf{Z}\|_* + \|\mathbf{L}\|_* + \frac{\lambda}{2} \|\mathbf{E}\|_F^2 + \langle \boldsymbol{\lambda}_1, \mathbf{1}^T \mathbf{Z} - \mathbf{1}^T \rangle + \frac{\beta}{2} \|\mathbf{1}^T \mathbf{Z} - \mathbf{1}^T\|^2 \\ & + \langle \boldsymbol{\lambda}_2, \mathbf{XZ} + \mathbf{LX} - \mathbf{X} - \mathbf{E} \rangle + \frac{\beta}{2} \|\mathbf{XZ} + \mathbf{LX} - \mathbf{X} - \mathbf{E}\|_F^2. \end{aligned}$$

Note that  $h(\mathbf{Z}, \mathbf{L}, \mathbf{E}) = \frac{1}{2} \|\mathbf{1}^T \mathbf{Z} - \mathbf{1}^T\|^2 + \frac{1}{2} \|\mathbf{XZ} + \mathbf{LX} - \mathbf{X} - \mathbf{E}\|_F^2$  is  $\{\eta_1 \mathbf{I}, \eta_2 \mathbf{I}, \eta_3 \mathbf{I}\}$ -smooth, where  $\eta_1 > \|\mathbf{1}\|^2 + 3\|\mathbf{X}\|_2^2$ ,  $\eta_2 > 3\|\mathbf{X}\|_2^2$  and  $\eta_3 > 3$ . By using such a property, L-ADMM-PS (3) solves (66) by the following updating rules

$$\begin{cases} \mathbf{Z}^{k+1} = \arg \min_{\mathbf{Z}} \|\mathbf{Z}\|_* + \frac{\beta^{(k)} \eta_1}{2} \left\| \mathbf{Z} - \mathbf{Z}^k + \frac{\mathbf{1} (\boldsymbol{\lambda}_1^k + \beta^{(k)} (\mathbf{1}^T \mathbf{Z}^k - \mathbf{1}^T)) + \mathbf{X}^T (\boldsymbol{\lambda}_2^k + \beta^{(k)} (\mathbf{XZ}^k + \mathbf{L}^k \mathbf{X} - \mathbf{X} - \mathbf{E}^k))}{\beta^{(k)} \eta_1} \right\|_F^2, \\ \mathbf{L}^{k+1} = \arg \min_{\mathbf{L}} \|\mathbf{L}\|_* + \frac{\beta^{(k)} \eta_2}{2} \left\| \mathbf{L} - \mathbf{L}^k + \frac{(\boldsymbol{\lambda}_2^k + \beta^{(k)} (\mathbf{XZ}^k + \mathbf{L}^k \mathbf{X} - \mathbf{X} - \mathbf{E}^k)) \mathbf{X}^T}{\beta^{(k)} \eta_2} \right\|_F^2, \\ \mathbf{E}^{k+1} = \arg \min_{\mathbf{E}} \frac{\lambda}{2} \|\mathbf{E}\|_F^2 - \langle \boldsymbol{\lambda}_2^k + \beta^{(k)} (\mathbf{XZ}^k + \mathbf{L}^k \mathbf{X} - \mathbf{X} - \mathbf{E}^k), \mathbf{E} \rangle + \frac{\beta^{(k)} \eta_3}{2} \|\mathbf{E} - \mathbf{E}^k\|_F^2, \\ \boldsymbol{\lambda}_1^{k+1} = \boldsymbol{\lambda}_1^k + \beta^{(k)} (\mathbf{1}^T \mathbf{Z}^{k+1} - \mathbf{1}^T), \\ \boldsymbol{\lambda}_2^{k+1} = \boldsymbol{\lambda}_2^k + \beta^{(k)} (\mathbf{XZ}^{k+1} + \mathbf{L}^{k+1} \mathbf{X} - \mathbf{X} - \mathbf{E}^{k+1}), \end{cases}$$

(c) Solve (66) by M-ADMM (3)

M-ADMM (3) divides the variables  $\{\mathbf{Z}, \mathbf{L}, \mathbf{E}\}$  into two super blocks, i.e.,  $\{\mathbf{Z}\}$  and  $\{\mathbf{L}, \mathbf{E}\}$ . Then it solves (66) by the following updating rules

$$\begin{cases} \mathbf{Z}^{k+1} = \arg \min_{\mathbf{Z}} \|\mathbf{Z}\|_* + \frac{\beta^{(k)}\eta_1}{2} \left\| \mathbf{Z} - \mathbf{Z}^k + \frac{\mathbf{1}(\boldsymbol{\lambda}_1^k + \beta^{(k)}(\mathbf{1}^T \mathbf{Z}^k - \mathbf{1}^T)) + \mathbf{X}^T(\boldsymbol{\lambda}_2^k + \beta^{(k)}(\mathbf{X}\mathbf{Z}^k + \mathbf{L}^k \mathbf{X} - \mathbf{X} - \mathbf{E}^k))}{\beta^{(k)}\eta_1} \right\|_F^2, \\ \mathbf{L}^{k+1} = \arg \min_{\mathbf{L}} \|\mathbf{L}\|_* + \frac{\beta^{(k)}\eta_2}{2} \left\| \mathbf{L} - \mathbf{L}^k + \frac{(\boldsymbol{\lambda}_2^k + \beta^{(k)}(\mathbf{X}\mathbf{Z}^{k+1} + \mathbf{L}^k \mathbf{X} - \mathbf{X} - \mathbf{E}^k))\mathbf{X}^T}{\beta^{(k)}\eta_2} \right\|_F^2, \\ \mathbf{E}^{k+1} = \arg \min_{\mathbf{E}} \frac{\lambda}{2} \|\mathbf{E}\|_F^2 + \frac{\beta^{(k)}}{2} \left\| \mathbf{X}\mathbf{Z}^{k+1} + \mathbf{L}^k \mathbf{X} - \mathbf{X} - \mathbf{E} + \frac{\boldsymbol{\lambda}_2^k}{\beta^{(k)}} \right\|_F^2 + \frac{\beta^{(k)}\eta_3}{2} \|\mathbf{E} - \mathbf{E}^k\|_F^2, \\ \boldsymbol{\lambda}_1^{k+1} = \boldsymbol{\lambda}_1^k + \beta^{(k)}(\mathbf{1}^T \mathbf{Z}^{k+1} - \mathbf{1}^T), \\ \boldsymbol{\lambda}_2^{k+1} = \boldsymbol{\lambda}_2^k + \beta^{(k)}(\mathbf{X}\mathbf{Z}^{k+1} + \mathbf{L}^{k+1} \mathbf{X} - \mathbf{X} - \mathbf{E}^{k+1}), \end{cases}$$

where  $\eta_1 = \|\mathbf{1}\|^2 + \|\mathbf{X}\|_2^2$ ,  $\eta_2 > 2\|\mathbf{X}\|_2^2$  and  $\eta_3 > 1$ .

## 2.2 Nonnegative Matirx Completion

$$\min_{\mathbf{X}, \mathbf{E}} \|\mathbf{X}\|_* + \frac{\lambda}{2} \|\mathbf{E}\|^2, \text{ s.t. } \mathcal{P}_\Omega(\mathbf{X}) + \mathbf{E} = \mathbf{B}, \mathbf{X} \geq \mathbf{0}, \quad (67)$$

(a) L-ADMM-PS

Problem (67) is equivalent to (see (94) in [22])

$$\begin{aligned} \min_{\mathbf{X}, \mathbf{E}, \mathbf{Z}} \|\mathbf{X}\|_* + \frac{\lambda}{2} \|\mathbf{E}\|^2, \\ \text{s.t. } \mathcal{P}_\Omega(\mathbf{X}) + \mathbf{E} = \mathbf{B}, \mathbf{X} = \mathbf{Z}, \mathbf{Z} \geq \mathbf{0}. \end{aligned} \quad (68)$$

The partial augmented Lagrangian function is

$$\begin{aligned} \mathcal{L}(\mathbf{X}, \mathbf{E}, \mathbf{Z}, \beta) = \|\mathbf{X}\|_* + \frac{\lambda}{2} \|\mathbf{E}\|^2 + \langle \boldsymbol{\lambda}_1, \mathcal{P}_\Omega(\mathbf{X}) + \mathbf{E} - \mathbf{B} \rangle + \frac{\beta}{2} \|\mathcal{P}_\Omega(\mathbf{X}) + \mathbf{E} - \mathbf{B}\|^2 \\ + \langle \boldsymbol{\lambda}_2, \mathbf{X} - \mathbf{Z} \rangle + \frac{\beta}{2} \|\mathbf{X} - \mathbf{Z}\|^2. \end{aligned}$$

Then L-ADMM-PS solves (68) by the following updating rules

$$\begin{cases} \mathbf{X}^{k+1} = \arg \min_{\mathbf{X}} \|\mathbf{X}\|_* + \frac{\beta^{(k)}\eta_1}{2} \left\| \mathbf{X} - \mathbf{X}^k + \frac{\mathcal{P}_\Omega(\boldsymbol{\lambda}_1^k) + \boldsymbol{\lambda}_2^k + \beta^{(k)}\mathcal{P}_\Omega(\mathbf{X}^k + \mathbf{E}^k - \mathbf{B}) + \beta^{(k)}(\mathbf{X}^k - \mathbf{Z}^k)}{\beta^{(k)}\eta_1} \right\|_F^2, \\ \mathbf{E}^{k+1} = \arg \min_{\mathbf{E}} \frac{\lambda}{2} \|\mathbf{E}\|^2 + \langle \mathbf{E}, \boldsymbol{\lambda}_1^k + \beta^{(k)}(\mathcal{P}_\Omega(\mathbf{X}^k) + \mathbf{E}^k - \mathbf{B}) \rangle + \frac{\beta^{(k)}\eta_2}{2} \|\mathbf{E} - \mathbf{E}^k\|^2, \\ \mathbf{Z}^{k+1} = \arg \min_{\mathbf{Z} \geq \mathbf{0}} \langle \boldsymbol{\lambda}_2^k + \beta^{(k)}(\mathbf{X}^k - \mathbf{Z}^k), -\mathbf{Z} \rangle + \frac{\beta^{(k)}}{2} \|\mathbf{X}^k - \mathbf{Z}\|^2 + \frac{\beta^{(k)}\eta_3}{2} \|\mathbf{X}^k - \mathbf{Z}\|^2, \\ \boldsymbol{\lambda}_1^{k+1} = \boldsymbol{\lambda}_1^k + \beta^{(k)}(\mathcal{P}_\Omega(\mathbf{X}^{k+1}) + \mathbf{E}^{k+1} - \mathbf{B}), \\ \boldsymbol{\lambda}_2^{k+1} = \boldsymbol{\lambda}_2^k + \beta^{(k)}(\mathbf{X}^{k+1} - \mathbf{Z}^{k+1}), \end{cases}$$

where  $\eta_1 > 3 + 2$ ,  $\eta_2 > 3 + 2$  and  $\eta_3 > 2$ .

(b) M-ADMM

Problem (67) is equivalent to

$$\begin{aligned} \min_{\mathbf{X}, \mathbf{E}, \mathbf{Z}} \|\mathbf{X}\|_* + \frac{\lambda}{2} \|\mathbf{E}\|^2, \\ \text{s.t. } \mathcal{P}_\Omega(\mathbf{Z}) + \mathbf{E} = \mathbf{B}, \mathbf{X} = \mathbf{Z}, \mathbf{Z} \geq \mathbf{0}. \end{aligned} \quad (69)$$

The partial augmented Lagrangian function is

$$\begin{aligned} \mathcal{L}(\mathbf{X}, \mathbf{E}, \mathbf{Z}, \beta) = \|\mathbf{X}\|_* + \frac{\lambda}{2} \|\mathbf{E}\|^2 + \langle \boldsymbol{\lambda}_1, \mathcal{P}_\Omega(\mathbf{Z}) + \mathbf{E} - \mathbf{B} \rangle + \frac{\beta}{2} \|\mathcal{P}_\Omega(\mathbf{Z}) + \mathbf{E} - \mathbf{B}\|^2 \\ + \langle \boldsymbol{\lambda}_2, \mathbf{X} - \mathbf{Z} \rangle + \frac{\beta}{2} \|\mathbf{X} - \mathbf{Z}\|^2. \end{aligned}$$

Partition the three blocks into two super blocks  $\{\mathbf{X}, \mathbf{E}\}$  and  $\{\mathbf{Z}\}$ . Then M-ADMM solves (69) by the following updating rules

$$\begin{cases} \mathbf{X}^{k+1} = \arg \min_{\mathbf{X}} \|\mathbf{X}\|_* + \frac{\beta^{(k)}}{2} \left\| \mathbf{X} - \mathbf{Z}^k + \frac{\boldsymbol{\lambda}_2^k}{\beta^{(k)}} \right\|_F^2, \\ \mathbf{E}^{k+1} = \arg \min_{\mathbf{E}} \frac{\lambda}{2} \|\mathbf{E}\|^2 + \langle \boldsymbol{\lambda}_1^k, \mathbf{E} \rangle + \frac{\beta^{(k)}}{2} \|\mathcal{P}_\Omega(\mathbf{Z}^k) + \mathbf{E} - \mathbf{B}\|^2, \\ \mathbf{Z}^{k+1} = \arg \min_{\mathbf{Z} \geq 0} \langle \boldsymbol{\lambda}_1^k, \mathcal{P}_\Omega(\mathbf{Z}) \rangle + \frac{\beta^{(k)}}{2} \|\mathcal{P}_\Omega(\mathbf{Z}) + \mathbf{E}^{k+1} - \mathbf{B}\|^2 + \langle \boldsymbol{\lambda}_2^k, -\mathbf{Z} \rangle + \frac{\beta^{(k)}}{2} \|\mathbf{X}^{k+1} - \mathbf{Z}\|^2, \\ \boldsymbol{\lambda}_1^{k+1} = \boldsymbol{\lambda}_1^k + \beta^{(k)} (\mathcal{P}_\Omega(\mathbf{X}^{k+1}) + \mathbf{E}^{k+1} - \mathbf{B}), \\ \boldsymbol{\lambda}_2^{k+1} = \boldsymbol{\lambda}_2^k + \beta^{(k)} (\mathbf{X}^{k+1} - \mathbf{Z}^{k+1}). \end{cases}$$

Note that the  $\mathbf{Z}^{k+1}$  updating has a closed form solution.

### 3 A List of Problems Involved in Our Released Toolbox

Table 5 gives a list of convex problems in compressed sensing solved by M-ADMM in our released LibADMM package. For each problem, we consider its specific structure to implement efficient M-ADMM by using several techniques proposed in this work.

TABLE 5: Applicability of the LibADMM package

Model	Problem	Function	Description and Reference
Sparse models	$\min_{\mathbf{x}} r(\mathbf{x})$ s.t. $\mathbf{Ax} = \mathbf{b}$	$r(\mathbf{x}) = \ \mathbf{x}\ _1$	l1
		$r(\mathbf{x}) = \sum_{g \in \mathcal{G}} \ \mathbf{x}_g\ _2$	group11
		$r(\mathbf{x}) = \ \mathbf{x}\ _1 + \lambda_2 \ \mathbf{x}\ _2^2$	elasticnet
		$r(\mathbf{x}) = \ \mathbf{x}\ _1 + \lambda_2 \sum_{i=2}^p  x_i - x_{i-1} $	fused11
		$r(\mathbf{x}) = \ \mathbf{A} \text{Diag}(\mathbf{x})\ _*$	tracelasso
		$r(\mathbf{x}) = \frac{1}{2} \ \mathbf{x}\ _{\text{ksp}}^2$	k support
	$\min_{\mathbf{x}, \mathbf{e}} l(\mathbf{e}) + \lambda r(\mathbf{x})$ s.t. $\mathbf{Ax} + \mathbf{e} = \mathbf{b}$	$l(\mathbf{e}) = \ \mathbf{e}\ _1$	l1R
		$l(\mathbf{e}) = \ \mathbf{e}\ _1$	group11R
		$l(\mathbf{e}) = \ \mathbf{e}\ _1$	elasticnetR
		$l(\mathbf{e}) = \frac{1}{2} \ \mathbf{e}\ _2^2$	fused11R
		$l(\mathbf{e}) = \frac{1}{2} \ \mathbf{e}\ _2^2$	tracelassoR
		$l(\mathbf{e}) = \frac{1}{2} \ \mathbf{e}\ _2^2$	k supportR
Low-rank matrix models	$\min_{\mathbf{X}, \mathbf{E}} \ \mathbf{X}\ _* + \lambda l(\mathbf{E}), \text{ s.t. } \mathcal{P}_\Omega(\mathbf{X}) + \mathbf{E} = \mathcal{M}$	lrmcR	Reg. Low-rank matrix completion
	$\min_{\mathbf{X}, \mathbf{E}} \ \mathbf{X}\ _* + \lambda l(\mathbf{E}), \text{ s.t. } \mathbf{A} = \mathbf{BX} + \mathbf{E}$	lrr	Low-rank representation
	$\min_{\mathbf{Z}, \mathbf{L}, \mathbf{E}} \ \mathbf{Z}\ _* + \ \mathbf{L}\ _* + \lambda l(\mathbf{E})$ s.t. $\mathbf{XZ} + \mathbf{LX} - \mathbf{X} = \mathbf{E}$	latlrr	Latent low-rank representation
	$\min_{\mathbf{X}, \mathbf{E}} \ \mathbf{X}\ _* + \lambda_1 \ \mathbf{X}\ _1 + \lambda_2 l(\mathbf{E})$ s.t. $\mathbf{A} = \mathbf{BX} + \mathbf{E}$	lrsr	Low-rank and sparse representation
	$\min_{\mathbf{L}_i, \mathbf{S}_i} \ \mathbf{L}\ _* + \lambda \sum_{i=1}^m \ \mathbf{S}_i\ _1$ s.t. $\mathbf{X}_i = \mathbf{L} + \mathbf{S}_i, i = 1, \dots, m, \mathbf{L} \geq 0, \mathbf{L}\mathbf{1} = \mathbf{1}$	rmsc	Robust multi-view spectral clustering
	$\min_{\mathbf{Z}_i, \mathbf{E}_i} \sum_{i=1}^K (\ \mathbf{Z}_i\ _* + \lambda l(\mathbf{E}_i)) + \alpha \ \mathbf{Z}\ _{2,1}$ s.t. $\mathbf{X}_i = \mathbf{X}_i \mathbf{Z}_i + \mathbf{E}_i, i = 1, \dots, K$	mlap	Multi-task low-rank affinity pursuit
	$\min_{\mathbf{L}, \mathbf{S}} \ \mathbf{L}\ _* + \lambda \ \mathbf{C} \circ \mathbf{S}\ _1, \text{ s.t. } \mathbf{A} = \mathbf{L} + \mathbf{S}, 0 \leq \mathbf{L} \leq \mathbf{1}$	igc	Improved graph clustering
	$\min_{\mathbf{P}} \langle \mathbf{P}, \mathbf{L} \rangle + \lambda \ \mathbf{P}\ _1, \text{ s.t. } 0 \preceq \mathbf{P} \preceq \mathbf{I}, \text{Tr}(\mathbf{P}) = k$	sparsesc	Sparse spectral clustering
Low-rank tensor models	$\min_{\mathcal{L}, \mathcal{S}} \sum_{i=1}^k \alpha_i \ \mathcal{L}_{(i)}\ _* + \ \mathcal{S}\ _1, \text{ s.t. } \mathcal{X} = \mathcal{L} + \mathcal{S}$	trpca_snn	Tensor robust PCA based on sum of nuclear norm
	$\min_{\mathcal{X}} \sum_{i=1}^k \alpha_i \ \mathcal{X}_{(i)}\ _*, \text{ s.t. } \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{M})$	lrtc_snn	Low-rank tensor completion based on sum of nuclear norm
	$\min_{\mathcal{X}, \mathcal{E}} \sum_{i=1}^k \alpha_i \ \mathcal{X}_{(i)}\ _* + \lambda l(\mathcal{E})$ s.t. $\mathcal{P}_\Omega(\mathcal{X}) + \mathcal{E} = \mathcal{M}$	lrtcR_snn	Reg. low-rank tensor completion based on sum of nuclear norm
	$\min_{\mathcal{L}, \mathcal{S}} \ \mathcal{L}\ _* + \lambda \ \mathcal{S}\ _1, \text{ s.t. } \mathcal{X} = \mathcal{L} + \mathcal{S}$	trpca_tnn	Tensor Robust PCA based on tensor nuclear norm
	$\min_{\mathcal{X}} \ \mathcal{X}\ _*, \text{ s.t. } \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{M})$	lrtc_tnn	Low-rank tensor completion based on tensor nuclear norm
	$\min_{\mathcal{X}, \mathcal{E}} \ \mathcal{X}\ _* + \lambda l(\mathcal{E}), \text{ s.t. } \mathcal{P}_\Omega(\mathcal{X}) + \mathcal{E} = \mathcal{M}$	lrtcR_tnn	Reg. low-rank tensor completion based on tensor nuclear norm

\*In this table, the loss function  $l(\cdot)$  can be  $\|\cdot\|_1, \frac{1}{2} \|\cdot\|_F^2$  and  $\|\cdot\|_{2,1}$ . The  $\|\cdot\|_{2,1}$  norm is only applicable to the matrix.