Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.OA-2018-0108

Vol. **13**, No. 1, pp. 200-219
February 2020

A Fast Symmetric Alternating Direction Method of Multipliers

Gang Luo[1,*] and Qingzhi Yang[2,1]

[1] *School of Mathematical Sciences and LPMC, Nankai University, Tianjin 300071, China*
[2] *School of Mathematics and Statistics, Kashi University, Kashi 844006, China*

Received 17 October 2018; Accepted (in revised version) 20 May 2019

Abstract. In recent years, alternating direction method of multipliers (ADMM) and its variants are popular for the extensive use in image processing and statistical learning. A variant of ADMM: symmetric ADMM, which updates the Lagrange multiplier twice in one iteration, is always faster whenever it converges. In this paper, combined with Nesterov's accelerating strategy, an accelerated symmetric ADMM is proposed. We prove its $\mathcal{O}(\frac{1}{k^2})$ convergence rate under strongly convex condition. For the general situation, an accelerated method with a restart rule is proposed. Some preliminary numerical experiments show the efficiency of our algorithms.

AMS subject classifications: 90C25, 90C30, 49M29, 65B99
Key words: Nesterov's accelerating strategy, alternating direction method of multipliers, symmetric ADMM, separable linear constrained optimization.

1. Introduction

In this paper, we consider the following convex minimization problem with a linear constrain and a separable objective function:

$$\begin{cases} \min\limits_{x,y} f(x)+g(y) \\ \text{s.t. } Ax+By=c, \end{cases} \tag{1.1}$$

where A, B are linear maps and f, g are convex functions. Problem (1.1) has found numerous applications in statistic and image processing. The augment Lagrangian formulation of (1.1) is

$$\max_{\lambda}\min_{x,y} f(x)+g(y)-\langle\lambda, Ax+By-c\rangle+\frac{\rho}{2}\|Ax+By-c\|^2, \tag{1.2}$$

where λ is the dual variable and ρ is a penalty parameter. Solving (1.1) via (1.2) is exactly the augmented Lagrangian method by Hestenes [16] and Powell [22]. For

*Corresponding author. *Email address:* luogangnk@gmail.com (G. Luo)

the separable structure in object function in above class problem, alternating direction method of the multipliers (ADMM) algorithm [9], one variant of ALM, is preferred. It minimizes (1.2) on x, y alternatively, then updates the dual variable λ. Readers can refer to the review paper [2] for some applications of ADMM in statistics and machine learning fields.

The iterative scheme of ADMM on (1.1) is

$$\begin{cases} x^{k+1} = \operatorname{argmin}_x f(x) - \langle \lambda^k, Ax\rangle + \dfrac{\rho}{2}\|Ax + By^k - c\|^2, \\ y^{k+1} = \operatorname{argmin}_y g(y) - \langle \lambda^k, By\rangle + \dfrac{\rho}{2}\|Ax^{k+1} + By - c\|^2, \\ \lambda^{k+1} = \lambda^k - \rho(Ax^{k+1} + By^{k+1} - c), \end{cases} \tag{1.3}$$

where x, y and Lagrange multiplier λ are updated in each iteration. ADMM is shown to be equivalent to the Douglas-Rachford splitting method (DRSM) [5] on the dual problem of (1.1). The convergence of ADMM under general condition is guaranteed for two block situation, and a proof can be found in [2]. The above algorithm can be easily extended to solve linear constrained minimization with three or more separated block objective function, while in these cases, its convergence is no longer guaranteed under general conditions [4].

Apply another famous splitting method: Peaceman-Rachford splitting method (PRSM) [21] on the dual problem of (1.1), we get a variation of ADMM and its iterative scheme is

$$\begin{cases} x^{k+1} = \operatorname{argmin}_x f(x) - \langle \lambda^k, Ax\rangle + \dfrac{\rho}{2}\|Ax + By^k - c\|^2, \\ \lambda^{k+\frac{1}{2}} = \lambda^k - \rho(Ax^{k+1} + By^k - c), \\ y^{k+1} = \operatorname{argmin}_y g(y) - \langle \lambda^{k+\frac{1}{2}}, By\rangle + \dfrac{\rho}{2}\|Ax^{k+1} + By - c\|^2, \\ \lambda^{k+1} = \lambda^{k+\frac{1}{2}} - \rho(Ax^{k+1} + By^{k+1} - c). \end{cases} \tag{1.4}$$

This algorithm is called symmetric ADMM (sADMM) for it updates λ twice in one iteration.

Different from ADMM, symmtric ADMM requires more to ensure its convergence [12], but it shows a faster convergence than ADMM in numerical computing. In [12], a contractive step size $a \in (0, 1)$ was introduced to the dual variable updating step to ensure the convergence of the algorithm

$$\begin{cases} x^{k+1} = \operatorname{argmin}_x f(x) - \langle \lambda^k, Ax\rangle + \dfrac{\rho}{2}\|Ax + By^k - c\|^2, \\ \lambda^{k+\frac{1}{2}} = \lambda^k - a\rho(Ax^{k+1} + By^k - c), \\ y^{k+1} = \operatorname{argmin}_y g(y) - \langle \lambda^{k+\frac{1}{2}}, By\rangle + \dfrac{\rho}{2}\|Ax^{k+1} + By - c\|^2, \\ \lambda^{k+1} = \lambda^{k+\frac{1}{2}} - a\rho(Ax^{k+1} + By^{k+1} - c). \end{cases} \tag{1.5}$$

ADMM and sADMM are the first-order algorithms. In [12, 14, 15], their $\mathcal{O}(\frac{1}{k})$ convergences were established. In practice, they can converge slowly to reach a high accuracy,

this is unlike some high precision algorithms such as Newton's method or interior-point method [2]. A lot of numerical results have shown that these iterative schemes have a large tail that it will converge slowly after some iterations. There are some other variants of ADMM to improve computational performance. In [13], a varying penalty parameter ρ strategy was proposed. An over-relaxation scheme in $y-$ and λ-updates was analyzed in [7], where Ax^{k+1} was replaced by

$$a^k Ax^{k+1} - (1 - a^k)(By^k - c),$$

and in [6] and [8], experiments suggested that $a^k \in [1.5, 1.8]$ could improve the convergence of the algorithm. An inexact minimization of the subproblem can be carried out due to [7], this is especially important when the subproblems are not easy to solve exactly. Recently, Yue et al. gave a criterion of the inexact solver of subproblem in ADMM which guaranteed the convergence of the algorithm [25]. Some variants of ADMM involve performing $x-$, $y-$ and $\lambda-$ updates in a varying order or multiple times, see more in [24].

When $A = I$ or $B = I$, an accelerated primal dual algorithm was proposed in [3] to solve (1.1), a rate of convergence $\mathcal{O}(\frac{1}{k^2})$ for the primal dual gap was proved for problems with some regularity in the primal or dual objective and it is linearly convergent $(\mathcal{O}(\frac{1}{e^k}))$ for some smooth cases. When $A = B = I$, Goldfarb et al. [10] presented both basic and accelerated first-order alternating linearization algorithms for solving (1.1). In [20], Ouyang et al. incorporated a multi-step acceleration scheme into linearized ADMM and demonstrated a better convergence rate in terms of dependence on the Lipschitz constant of the smooth component. In [11], Goldstein et al. introduced the well known Nesterov's accelerating method [18] into ADMM algorithm and proved the $\mathcal{O}(\frac{1}{k^2})$ convergence rate on the dual of (1.1) for the problem with strongly convexity assumption of the primal objective. A accelerated ADMM with restart rule was also proposed for general convex condition. Later in [17], Kadkhodaie et al. relaxed the quadratic assumption on G in the accelerated ADMM [11] and proposed an accelerated ADMM called A2DM2.

Motivated by them, in this paper we try to incorporate Nesterov's accelerating strategy into the framework of symmetric ADMM. First we add the extrapolation step directly after the dual variable updating of symmetric ADMM as stated in Algorithm 3.1. A $\mathcal{O}(\frac{1}{k^2})$ convergence rate on dual objective is achieved with restriction on parameter ρ and strongly convexity condition of the objection function. Then upon on the research of symmetric ADMM with contractive step size in [12], an accelerated symmetric ADMM with restart rule is presented and we show its monotonic convergence. A double dual variable updating in one iteration and the flexibility of choosing contractive step size in our algorithms help a fast convergence over the fast ADMM in [11] as is demonstrated by some preliminary numerical experiment at last.

The rest paper is organized as follows. In Section 2, we introduce Nesterov's accelerating strategy in brief. In Section 3, we present an accelerated symmetric ADMM algorithm and prove its $\mathcal{O}(\frac{1}{k^2})$ convergence rate on dual optimal under strongly convex

condition. In Section 4, a monotonous convergence algorithm with a restart rule is given. Some preliminary numerical results are presented in Section 5.

Notation 1.1. $\|\cdot\|_2$ denotes the Euclidean norm for vector or spectral norm for matrix. $|\cdot|$ denotes the l_1 norm for vector.

2. Nesterov's accelerating strategy

In a seminal paper [18], Nesterov showed that a special extrapolation on sequence generated by gradient method could results in dramatic change on convergence speed of simplest gradient method: from $\mathcal{O}(\frac{1}{k})$ to $\mathcal{O}(\frac{1}{k^2})$. The new algorithm uses the information in the previous two iterations to generate the next iteration point. Here we introduce the basic framework briefly. Consider an unconstrained convex minimization

$$\min_{x\in\mathbb{R}^n} f(x),$$

where f has a Lipschitz continuous gradient, i.e., there exists a constant number $L_{\nabla f} > 0$ such that

$$\|\nabla f(x_1) - \nabla f(x_2)\|_2 \le L_{\nabla f}\|x_1 - x_2\|_2, \qquad \forall x_1, x_2 \in \mathbb{R}^n,$$

$\nabla f(x)$ denotes the gradient of f at x. Nesterov proposed following algorithm to minimize the above problem.

Algorithm 2.1 Nesterov's accelerated gradient method.

Require: $\alpha_0 = 1$, $x_0 = y_1 \in \mathbb{R}^N$, $\tau < \frac{1}{L(\nabla f)}$

1: **for** $k = 1, 2, 3, \cdots$ **do**
2: $\quad x_k = y_k - \tau\nabla f(y_k)$
3: $\quad \alpha_{k+1} = \frac{1+\sqrt{4\alpha_k^2+1}}{2}$
4: $\quad y_{k+1} = x_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(x_k - x_{k-1})$
5: **end for**

An extrapolation step is added with coefficient inside changing via iteration. It is proved that Algorithm 2.1 can reach $\mathcal{O}(\frac{1}{k^2})$ convergence rate on objective with the smooth assumption.

Theorem 2.1. *$\{x_k\}$ is the sequence generated by Algorithm* 2.1 *and x^* is one of the minimization of $f(x)$, then*

$$f(x_k) - f(x^*) \le \frac{2\|x_0 - x^*\|^2}{\tau(k+1)^2}. \tag{2.1}$$

The proof can be found in [18]. Based on Nesterov's accelerating strategy, a lot of accelerated versions of first-order algorithms have been proposed for various optimization problems in the literatures, e.g., [1, 10].

3. Accelerated sADMM

In this section, we combine Nesterov's accelerating strategy with sADMM, and propose an accelerated sADMM as below.

Steps 6-8 are introduced into the original algorithm, specifically the extrapolation is upon on dual variable λ and primal variable y. In the accelerated ADMM [11], the extrapolation on y is

$$\hat{y}^{k+1} = y^{k+1} + \frac{\theta_{k+1}(1-\theta_k)}{\theta_k}(y^{k+1} - y^k).$$

Later in A2DM2 [17], the extrapolation on y takes the form

$$\hat{y}^{k+1} = \mathrm{argmin}_y g(y) - \langle By, \hat{\lambda}^{k+1}\rangle,$$

which eliminates the quadratic requirement on $g(y)$ in [11]. When $g(y)$ is quadratic, one can easily verify that the two extrapolations are equivalent. In the rest of this section, we demonstrate a $\mathcal{O}(\frac{1}{k^2})$ convergence rate on dual of (1.1) with the condition that $f(x)$ and $g(y)$ are strongly convex functions.

We assume the strong convexity of F with modules σ_F such that for any x_1, x_2,

$$F(x_1) - F(x_2) \geq \langle v, x_1 - x_2\rangle + \frac{\sigma_f}{2}\|x_1 - x_2\|^2, \qquad \forall v \in \partial F(x_2),$$

where $\partial F(\cdot)$ denotes the subdifferential set of F. For a convex function F, its Fenchel conjugate function F^* is defined as

$$F^*(\lambda) := \sup_z\{\langle z, \lambda\rangle - F(z)\}.$$

It is well known that

$$\lambda \in \partial F(z) \iff z \in \partial F^*(\lambda).$$

Algorithm 3.1 Accelerated sADMM.

Require: $\theta_0 = \theta_1 = 1$, $y^1 = y^0$, $\hat{\lambda}^1 = \lambda^1 = \lambda^0$ and $B^T\lambda^1 \in \partial g(y^1)$

1: **for** $k = 1, 2, 3, \cdots$ **do**
2: $x^{k+1} = \mathrm{argmin}_x f(x) + \langle\hat{\lambda}^k, -Ax\rangle + \frac{\rho}{2}\|Ax + B\hat{y}^k - c\|^2$
3: $\lambda^{k+\frac{1}{2}} = \hat{\lambda}^k - \rho(Ax^{k+1} + B\hat{y}^k - c)$
4: $y^{k+1} = \mathrm{argmin}_y g(y) + \langle\lambda^{k+\frac{1}{2}}, -By\rangle + \frac{\rho}{2}\|Ax^{k+1} + By - c\|^2$
5: $\lambda^{k+1} = \lambda^{k+\frac{1}{2}} - \rho(Ax^{k+1} + By^{k+1} - c)$
6: $\theta_{k+1} = \frac{2}{k+1}$
7: $\hat{\lambda}^{k+1} = \lambda^{k+1} + \frac{\theta_{k+1}(1-\theta_k)}{\theta_k}(\lambda^{k+1} - \lambda^k)$
8: $\hat{y}^{k+1} = \mathrm{argmin}_y g(y) - \langle By, \hat{\lambda}^{k+1}\rangle$
9: **end for**

For a strongly convex function F with modulus $\sigma_F > 0$, its conjugate function F^* has a Lipschitz continuous gradient with $L_{\nabla F^*} = \frac{1}{\sigma_F}$. Let $f^*(\lambda)$ and $g^*(y)$ be the conjugate functions of f and g respectively. Since both f and g are strongly convex functions, then $f^*(\lambda)$ and $g^*(\lambda)$ have Lipschitz continuous gradients with Lipschitz constant $L_{\nabla f^*} = \frac{1}{\sigma_f}$ and $L_{\nabla g^*} = \frac{1}{\sigma_g}$, respectively.

The dual function of (1.1) is

$$\begin{aligned} d(\lambda) :=& \min_{x,y}\{f(x) + g(y) + \langle \lambda, Ax + By - c\rangle\} \\ =& - f^*(-A^T\lambda) - g^*(-B^T\lambda) - \langle \lambda, c\rangle. \end{aligned} \tag{3.1}$$

With strong duality holds, solving Problem (1.1) is equivalent to maximizing the dual function $d(\lambda)$, which is equivalent to minimizing $p(\lambda) := f^*(A^T\lambda) + g^*(B^T\lambda) - \langle \lambda, c\rangle$. From the strongly convex assumption, $p(\lambda)$ has a Lipschitz continuous gradient, which satisfies the condition of Nesterov's accelerating strategy. Nextly we shall research deeply into Algorithm 3.1 and prove that incorporation of extrapolation step into sADMM can lead a fast convergence on $p(\lambda)$.

The first order optimal condition of the first subproblem in Algorithm 3.1 is

$$0 \in \partial f(x^{k+1}) - A^T\hat{\lambda}^k + \rho A^T(Ax^{k+1} + B\hat{y}^k - c),$$

i.e.,

$$A^T\lambda^{k+\frac{1}{2}} \in \partial f(x^{k+1}).$$

Similarly, we have

$$B^T\lambda^{k+1} \in \partial g(y^{k+1}).$$

From the property of conjugate function, we have

$$\nabla f^*(A^T\lambda^{k+\frac{1}{2}}) = x^{k+1}, \qquad \nabla g^*(B^T\lambda^{k+1}) = y^{k+1}. \tag{3.2}$$

The sequence generated by Algorithm 3.1 satisfies the following lemmas.

Lemma 3.1. *Assume $\nabla g^*(B^T\hat{\lambda}^k) = \hat{y}^k$ and $\rho \le \frac{1}{L_{\nabla g^*}\|B\|_2^2}$, $\lambda^{k+\frac{1}{2}}$ is generated from Algorithm* 3.1*, we have for any λ,*

$$p(\lambda^{k+\frac{1}{2}}) - p(\lambda) \le \frac{1}{2\rho}(\|\hat{\lambda}^k - \lambda\|^2 - \|\lambda^{k+\frac{1}{2}} - \lambda\|^2).$$

Proof. By Lipschitz continuous of the gradient of g^*, one has

$$g^*(B^T\lambda^{k+\frac{1}{2}}) \le g^*(B^T\hat{\lambda}^k) + \langle B\nabla g^*(B^T\hat{\lambda}^k), \lambda^{k+\frac{1}{2}} - \hat{\lambda}^k\rangle + \frac{L_{\nabla g^*}}{2}\|B^T(\lambda^{k+\frac{1}{2}} - \hat{\lambda}^k)\|^2,$$

and for the convexity of f^* and g^*, we have

$$f^*(A^T\lambda) - f^*(A^T\lambda^{k+\frac{1}{2}}) \geq \langle A\nabla f^*(A^T\lambda^{k+\frac{1}{2}}), \lambda - \lambda^{k+\frac{1}{2}}\rangle,$$
$$g^*(B^T\lambda) - g^*(B^T\hat{\lambda}^k) \geq \langle B\nabla g^*(B^T\hat{\lambda}^k), \lambda - \hat{\lambda}^k\rangle.$$

Add the above two inequalities together. For $p(\lambda) = f^*(A^T\lambda) + g^*(B^T\lambda) - \langle\lambda, c\rangle$, we have

$$\begin{aligned}
&p(\lambda^{k+\frac{1}{2}}) - p(\lambda)\\
\leq&\langle A\nabla f^*(A^T\lambda^{k+\frac{1}{2}}) + B\nabla g^*(B^T\hat{\lambda}^k) - c, \lambda^{k+\frac{1}{2}} - \lambda\rangle + \frac{L_{\nabla g^*}}{2}\|B^T(\lambda^{k+\frac{1}{2}} - \hat{\lambda}^k)\|^2\\
\leq&\langle Ax^{k+1} + B\hat{y}^k - c, \lambda^{k+\frac{1}{2}} - \lambda\rangle + \frac{L_{\nabla g^*}\|B\|_2^2}{2}\|\lambda^{k+\frac{1}{2}} - \hat{\lambda}^k\|^2\\
=&\frac{1}{\rho}\langle\hat{\lambda}^k - \lambda^{k+\frac{1}{2}}, \lambda^{k+\frac{1}{2}} - \lambda\rangle + \frac{L_{\nabla g^*}\|B\|_2^2}{2}\|\lambda^{k+\frac{1}{2}} - \hat{\lambda}^k\|^2\\
=&\frac{1}{2\rho}(\|\hat{\lambda}^k - \lambda\|^2 - \|\lambda^{k+\frac{1}{2}} - \lambda\|^2) + \frac{1}{2}\Big(L_{\nabla g^*}\|B\|_2^2 - \frac{1}{\rho}\|\lambda^{k+\frac{1}{2}} - \hat{\lambda}^k\|^2\Big)\\
\leq&\frac{1}{2\rho}(\|\hat{\lambda}^k - \lambda\|^2 - \|\lambda^{k+\frac{1}{2}} - \lambda\|^2). \qquad (3.3)
\end{aligned}$$

The second inequality follows from (3.2). The second equality follows from the equality: $\langle a-b, b-c\rangle = \frac{1}{2}(\|a-c\|^2 - \|b-c\|^2 - \|a-b\|^2)$. The last inequality follows from the assumption of the lemma. The proof is completed. □

Similarly, we have the following lemma.

Lemma 3.2. *Assume* $\nabla f^*(A^T\lambda^{k+\frac{1}{2}}) = x^{k+1}$ *and* $\rho \leq \frac{1}{L_{\nabla g^*}\|A\|_2^2}$, λ^{k+1} *is generated from Algorithm* 3.1*, then for any* λ*, we have*

$$p(\lambda^{k+1}) - p(\lambda) \leq \frac{1}{2\rho}(\|\lambda^{k+\frac{1}{2}} - \lambda\|^2 - \|\lambda^{k+1} - \lambda\|^2).$$

The assumption $\nabla g^*(B^\top\hat{\lambda}^k) = \hat{y}^k$ in Lemma 3.1 is satisfied for step 8 in Algorithm 3.1 and in Lemma 3.2 the assumption $\nabla f^*(A^\top\lambda^{k+\frac{1}{2}}) = x^{k+1}$ is established from (3.2). Let $\lambda = \lambda^{k+\frac{1}{2}}$ in Lemma 3.2, we have

$$p(\lambda^{k+1}) - p(\lambda^{k+\frac{1}{2}}) \leq \frac{1}{2\rho}(-\|\lambda^{k+1} - \lambda^{k+\frac{1}{2}}\|^2) \leq 0.$$

Combine Lemma 3.1 and Lemma 3.2, we have

$$p(\lambda^{k+\frac{1}{2}}) + p(\lambda^{k+1}) - 2p(\lambda) \leq \frac{1}{2\rho}(\|\hat{\lambda}^k - \lambda\|^2 - \|\lambda^{k+1} - \lambda\|^2),$$

then

$$p(\lambda^{k+1}) - p(\lambda) \leq \frac{1}{4\rho}(\|\hat{\lambda}^k - \lambda\|^2 - \|\lambda^{k+1} - \lambda\|^2). \qquad (3.4)$$

This inequality is fatal in proving the $\mathcal{O}(\frac{1}{k^2})$ convergence rate of the accelerated sADMM.

Theorem 3.1. *Assume* $\rho \le \min\{\frac{1}{L_{\nabla g^*}\|A\|_2^2}, \frac{1}{L_{\nabla f^*}\|B\|_2^2}\}$*, the sequence* $\{\lambda^k\}$ *is generated by Algorithm* 3.1*,* λ^* *is any minimization of* $p(\lambda)$*, then*

$$p(\lambda^{k+1}) - p(\lambda^*) \le \frac{1}{\rho(k+1)^2}\|\lambda^1 - \lambda^*\|^2, \qquad k \ge 1. \tag{3.5}$$

Proof. Set $\lambda = (1-\theta_k)\lambda^k + \theta_k\lambda^*$, $\theta_k \in (0,1)$. From the convexity of $p(\lambda)$, we have

$$\begin{aligned} p(\lambda^{k+1}) - p(\lambda) \ge& p(\lambda^{k+1}) - (1-\theta_k)p(\lambda^k) - \theta_k p(\lambda^*) \\ =& p(\lambda^{k+1}) - p(\lambda^*) - (1-\theta_k)(p(\lambda^k) - p(\lambda^*)). \end{aligned} \tag{3.6}$$

The right side of (3.4) is equivalent to

$$\begin{aligned} &\frac{1}{4\rho}(\|\hat{\lambda}^k - (1-\theta_k)\lambda^k - \theta_k\lambda^*\|^2 - \|\lambda^{k+1} - (1-\theta_k)\lambda^k - \theta_k\lambda^*\|^2) \\ =&\frac{\theta_k^2}{4\rho}\Big(\Big\|\frac{\hat{\lambda}^k}{\theta_k} - \frac{1-\theta_k}{\theta_k}\lambda^k - \lambda^*\Big\|^2 - \Big\|\frac{\lambda^{k+1}}{\theta_k} - \frac{1-\theta_k}{\theta_k}\lambda^k - \lambda^*\Big\|^2\Big). \end{aligned} \tag{3.7}$$

Denote $u^k := \frac{\lambda^k}{\theta_{k-1}} - \frac{1-\theta_{k-1}}{\theta_{k-1}}\lambda^{k-1}$. From Algorithm 3.1, we have

$$\begin{aligned} \frac{\hat{\lambda}^k}{\theta_k} - \frac{1-\theta_k}{\theta_k}\lambda^k =& \frac{1}{\theta_k}\Big(\lambda^k + \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}(\lambda^k - \lambda^{k-1})\Big) - \frac{1-\theta_k}{\theta_k}\lambda^k \\ =& \frac{\lambda^k}{\theta_{k-1}} - \frac{1-\theta_{k-1}}{\theta_{k-1}}\lambda^{k-1} = u^k. \end{aligned} \tag{3.8}$$

Then the right side of (3.4) becomes $\frac{\theta_k^2}{4\rho}(\|u^k - \lambda^*\|^2 - \|u^{k+1} - \lambda^*\|^2)$. Combine (3.4) and (3.6), we have

$$\begin{aligned} &\frac{1}{\theta_k^2}(p(\lambda^{k+1}) - p(\lambda^*)) + \frac{1}{4\rho}\|u^{k+1} - \lambda^*\|^2 \\ \le& \frac{1-\theta_k}{\theta_k^2}(p(\lambda^k) - p(\lambda^*)) + \frac{1}{4\rho}\|u^k - \lambda^*\|^2. \end{aligned} \tag{3.9}$$

In Algorithm 3.1, $\theta_k = \frac{2}{k+1}$ for $k \ge 1$, it is easy to verify that $\frac{1-\theta_k}{\theta_k^2} \le \frac{1}{\theta_{k-1}^2}$, so the above inequality becomes

$$\begin{aligned} &\frac{1}{\theta_k^2}(p(\lambda^{k+1}) - p(\lambda^*)) + \frac{1}{4\rho}\|u^{k+1} - \lambda^*\|^2 \\ \le& \frac{1}{\theta_{k-1}^2}(p(\lambda^k) - p(\lambda^*)) + \frac{1}{4\rho}\|u^k - \lambda^*\|^2 \cdots \\ \le& \frac{1}{\theta_1^2}(p(\lambda^2) - p(\lambda^*)) + \frac{1}{4\rho}\|u^2 - \lambda^*\|^2 \\ \le& \frac{1-\theta_1}{\theta_1^2}(p(\lambda^1) - p(\lambda^*)) + \frac{1}{4\rho}\|u^1 - \lambda^*\|^2 \\ =& \frac{1}{4\rho}\|u^1 - \lambda^*\|^2 = \frac{1}{4\rho}\|\lambda^1 - \lambda^*\|^2. \end{aligned} \tag{3.10}$$

Thus we finally get

$$p(\lambda^{k+1}) - p(\lambda^*) \leq \theta_k^2\Big(\frac{1}{4\rho}\|\lambda^1 - \lambda^*\|^2\Big) \leq \frac{1}{\rho(k+1)^2}\|\lambda^1 - \lambda^*\|^2.$$

Thus, we complete the proof. □

Remark 3.1. The choice of θ_k is not unique. One of the key part in the proof of the $\mathcal{O}(\frac{1}{k^2})$ convergence rate is the inequality $\frac{1-\theta_k}{\theta_k^2} \leq \frac{1}{\theta_{k-1}^2}$, so there are different strategies of choosing $\{\theta_k\}$. For example, let

$$\frac{1-\theta_k}{\theta_k^2} = \frac{1}{\theta_{k-1}^2},$$

i.e.,

$$\theta_k = \frac{\theta_{k-1}\big(\sqrt{\theta_{k-1}^2 + 4} - \theta_{k-1}\big)}{2},$$

and this is exactly the coefficients in Nestrov's accelerating strategy. Here $\theta_k = \frac{2}{k+1}$ in our algorithm also satisfies this inequality. In [19], the authors took a deeper look into the general form: $\theta_{k+1}^2 = (1-\theta_{k+1})\theta_k^2 + q\theta_{k+1}$, $q \in [0,1]$ and showed the choice of q greatly affected the speed of convergence even in quadratic programming.

Remark 3.2. The condition $B^T\lambda^1 \in \partial g(y^1)$ in the algorithm can be easily achieved by solving a y-subproblem and taking the solution as the initial point of the algorithm. An extra y-subproblem in the algorithm is needed to ensure $B^T\hat{\lambda}^k \in \partial g(\hat{y}^k)$. If $g(y)$ is a quadratic function, which is satisfied in many applications, this step can be replaced by

$$\hat{y}^{k+1} = y^{k+1} + \frac{\theta_{k+1}(1-\theta_k)}{\theta_k}(y^{k+1} - y^k).$$

Although we have proven the $\mathcal{O}(\frac{1}{k^2})$ convergence rate under strongly convex condition, we have to admit that this requirement is too restrict to some real applications, for example in lasso problem, or TV denoising problem, the object function has a non strongly convex term. In the next section, a restart version of the accelerated sADMM is proposed and it is applicable to these problems.

4. Accelerated sADMM with a restart rule

Since sADMM is not necessary to converge under general conditions, in [12] He et al. proposed a modified version of sADMM with a contractive step size and proved its convergence. In this section, we combine Nesterov's accelerating strategy with contractive step size strategy in a new algorithm as below.

Algorithm 4.1 Accelerated sADMM with a restart rule.

Require: $\theta_0 = \theta_1 = 1$, $y^1 = y^0$, $\hat{\lambda}^1 = \lambda^1 = \lambda^0$ and $B^T\lambda^1 \in \partial g(y^1)$, $a \in (0,1)$, $\eta = 0.99$

1: **for** $k = 1, 2, 3, \cdots$ **do**
2: $x^{k+1} = \mathrm{argmin}_x f(x) + \langle \hat{\lambda}^k, -Ax\rangle + \frac{\rho}{2}\|Ax + B\hat{y}^k - c\|^2$
3: $\lambda^{k+\frac{1}{2}} = \hat{\lambda}^k - a\rho(Ax^{k+1} + B\hat{y}^k - c)$
4: $y^{k+1} = \mathrm{argmin}_y g(y) + \langle \lambda^{k+\frac{1}{2}}, -By\rangle + \frac{\rho}{2}\|Ax^{k+1} + By - c\|^2$
5: $\lambda^{k+1} = \lambda^{k+\frac{1}{2}} - a\rho(Ax^{k+1} + By^{k+1} - c)$
6: $c_{k+1} = \|v^{k+1} - \hat{v}^k\|_H^2$
7: **if** $c_{k+1} \le \eta c_k$ **then**
8: $\theta_{k+1} = \frac{\theta_k(\sqrt{\theta_k^2+4} - \theta_k)}{2}$
9: $\hat{\lambda}^{k+1} = \lambda^{k+1} + \frac{\theta_{k+1}(1-\theta_k)}{\theta_k}(\lambda^{k+1} - \lambda^k)$
10: $\hat{y}^{k+1} = \mathrm{argmin}_y g(y) - \langle By, \hat{\lambda}^{k+1}\rangle$
11: **else**
12: $\theta_{k+1} = 1, \hat{y}^{k+1} = y^k, \hat{\lambda}^{k+1} = \lambda^k$
13: $c_{k+1} \leftarrow \eta^{-1} c_k$
14: **end if**
15: **end for**

In this algorithm, v^k and $\hat{v}^k$ are defined as $v^k := (y^k, \lambda^k)$, $\hat{v}^k := (\hat{y}^k, \hat{\lambda}^k)$. H is a positive semidefinite matrix:

$$H := \frac{1}{2}\begin{pmatrix} (2-a)\rho B^T B & -B^T \\ -B & \frac{1}{a\rho} I \end{pmatrix}. \tag{4.1}$$

For any vector a, we denote $\|a\|_H^2 := a^\top H a$. Restart process will take place whenever $c_{k+1} > \eta c_k$, otherwise extrapolations on λ and y execute. We assume that the step 2, 4, 10 in this algorithm have finite solutions through the iterations. This algorithm is motivated by the accelerated ADMM for weakly convex problem proposed in [11] and there exist three differences between them: Algorithm 4.1 has a double λ-update procedure, a contractive step size a and the restart criterion c_k here is different from which in [11]. Same to the accelerated ADMM for weakly convex problem, Algorithm 4.1 is guaranteed to converge for general convex problems.

Theorem 4.1. *Algorithm* 4.1 *converges in the sense that*

$$\lim_{k\to\infty} c_k = 0,$$

when both f and g are convex functions.

The keys of proving Theorem 4.1 are the two lemmas proposed in [12] and for the similarity of the proof to the accelerated ADMM for weakly convex problem in [11], we omit the detailed proof and only list the key lemmas from [12].

Lemma 4.1. *The sequence* $\{v^k\}$ *generated by the scheme* (1.5) *satisfies*

$$\|v^k - v^{k+1}\|_H^2 \leq \|v^{k-1} - v^k\|_H^2.$$

Lemma 4.2. *The sequence* $\{v^k\}$ *generated by the scheme* (1.5) *satisfies*

$$\|v^k - v^{k+1}\|_H^2 \leq \frac{2(1+a)}{(k+1)(1-a)}\|v^0 - v^*\|_H^2,$$

where $v^* = (y^*, \lambda^*)$ *is any optimal solution of the problem* (1.1) *and related dual optimal solution.*

For the reason of restart rule, here we adapt another choice of θ_k which is different from the choice in Algorithm 3.1. Although we can only ensure the convergence of Algorithm 4.1 under general convex conditions, not the $\mathcal{O}(\frac{1}{k^2})$ convergence rate for Algorithm 3.1 under strongly convex condition, Algorithm 4.1 usually achieves a better convergence behavior compared to Algorithm 3.1 and accelerated ADMM in [11] as shown in numerical experiments. This is perhaps for the double updating of dual variable λ and the flexibility of choosing constracitve step size a.

5. Numerical results

In this section, we give some preliminary numerical experiments to verify our algorithms under both strongly convex condition and weakly convex condition. We first consider the elastic net regularization problem which satisfies the strongly convex condition, then consider a weakly convex problem from image denoising. The codes run on Matlab R2013b installed on a laptop computer with 1.8GHz, i5-processor and 4G memory.

5.1. Elastic net regularization

To fit a large number of variables using a relatively small or noisy data set, a regularization term is introduced in statistics learning. Besides the least absolute shrinkage and selection operator (LASSO), i.e., l_1 regularization, elastic net regularization [26] is also popular. For the real world data, a simulation study shows that the elastic net model often outperforms the lasso, while keeps a similar sparsity of the representation. The model of the variable selection via elastic net regularization is

$$\min_x e_1|x| + \frac{e_2}{2}\|x\|^2 + \frac{1}{2}\|Mx - f\|^2, \tag{5.1}$$

where M represents the parameters input, f is the observation, x is the variable to predict. Model (5.1) can be reformulated as

$$\begin{cases} \min\limits_{x,y} e_1|x| + \dfrac{e_2}{2}\|x\|^2 + \dfrac{1}{2}\|My - f\|^2 \\ \text{s.t. } x - y = 0. \end{cases} \tag{5.2}$$

The two subproblems are involved in general ADMM form algorithms:

x-subproblem :

$$\min_x e_1|x| + \frac{e_2}{2}\|x\|^2 - x^\top \lambda^k + \frac{\rho}{2}\|x - y^k\|^2.$$

y-subproblem : solving linear equation

$$(M^\top M + \rho I)y = M^T f - \lambda^k + \rho x_k.$$

For the coefficient matrix $M^\top M + \rho I$ in y-subproblem is fixed, we can decompose it as $R^\top R$ in advance, where R is an upper triangular matrix. x-subproblem can be easily solved by a shrink operator and the solution is

$$x = \begin{cases} \dfrac{\lambda^k + \rho y^k - e_1}{e_2 + \rho}, & \lambda^k + \rho y^k \geq e_1, \\ 0, & -e_1 \leq \lambda^k + \rho y^k \leq e_1, \\ \dfrac{\lambda^k + \rho y^k + e_1}{e_2 + \rho}, & \lambda^k + \rho y^k \leq -e_1. \end{cases} \tag{5.3}$$

The elastic net regularization satisfies the strongly convex condition if M in (5.1) has full column rank. Here we use an example from Zou and Hastie [26]: Choose

$$x = (\underbrace{3, \cdots, 3}_{15}, \underbrace{0, \cdots, 0}_{25})$$

as the parameter to predict and the matrix $M \in \mathbb{R}^{50\times 40}$ is generated by

$$\begin{aligned} &M_i = Z_1 + e_i, \quad Z_1 \sim N(0,1), \quad i = 1, \cdots, 5, \\ &M_i = Z_2 + e_i, \quad Z_2 \sim N(0,1), \quad i = 6, \cdots, 10, \\ &M_i = Z_3 + e_i, \quad Z_3 \sim N(0,1), \quad i = 11, \cdots, 15, \\ &M_i \sim N(0,1), \quad M_i \text{ independent identically distributed}, \quad i = 16, \cdots, 40, \end{aligned}$$

where e_i are independent identically distributed $N(0, \sigma_e)$ vectors of length 50, $i = 1, \cdots, 15$ and Z_i are three random normal vectors of length 50, $i = 1, 2, 3$. A noisy measurement is added that $f = Mu + \eta$, where η is normally distributed with standard deviation 0.1. When the standard deviation of e_i changes, the conditional number of the matrix M changes, so we test two situations: (1) when $\sigma_e = 1$, which is suggested in [26], conditional number of M is round 20. In this case, the problem is regarded as well conditioned; (2) $\sigma_e = 0.1$, the correlations between first 15 columns of M is strong and the condition number of M is round 150. In this case, the problem is regarded as poor conditioned.

To test the behavior of the algorithms present in this paper, we compare them to the algorithms in [11], i.e., fast (accelerated) ADMM (Fadmm) and fast (accelerated) ADMM with restart rule (Fadmm + restart). Same to [11], we use the dual energy

gap (i.e., $d(\lambda^*) - d(\lambda^k)$) as the criterion for the quality of the solution and set the regularization parameters $e_1 = e_2 = 1$. From Theorem 3.1, the theoretical upper bound of ρ for fast symmetric ADMM is $\min\{e_2, \lambda_{smallest}(M^\top M)\}$, where $\lambda_{smallest}(M^\top M)$ is the smallest eigenvalue of $M^\top M$. The theoretical upper bound of ρ for fast ADMM (see [11, Theorem 1]) is $\sqrt[3]{e_2(\lambda_{smallest}(M^\top M))^2}$. Under the $e_2 = 1$ setting, we see that the theoretical upper bound of ρ for fast sADMM is alway smaller than the counterpart for fast ADMM except when $\lambda_{smallest}(M^\top M) = 1$. From numerical experiments, we find that the two theoretical upper bounds are too restrict for good behaviors of both algorithms and we test the algorithms with different ρ.

Figs. 1 and 2 show the convergence behavior of different algorithms on well conditioned and poor conditioned elastic net problem respectively. It is shown from (a) and (b) that large penalty parameter ρ leads to instability for both Fadmm and Fsadmm (Fsadmm is short for Algorithm 4.1) and ρ is stricter for Fsadmm than for Fadmm. This is in agreement with the requirement in Theorem 3.1. For Fadmm, the parameter ρ with best convergence performance on elastic net problem in poor conditioned situation is larger than that in well condition situation, while ρ in Fsadmm is not that sensitive. Restart rule works for both Fadmm and Fsadmm from (c) and (d). (e) and (f) show that the contractive step size a in Algorithms 3.1, 4.1 improves the convergence behavior, especial when ρ is large. In Fig. 2(e), Algorithm 3.1 dose not converge with $\rho = 4$, when a step size $a \in (0, 1)$ is added, the algorithm converges, though we can not prove the convergence of Algorithm 3.1 with a constrictive step size in theory. We compare the convergence behavior of different algorithms with best turned parameters on elastic net problem in Fig. 3. The best penalty parameter ρ for different algorithms is not same and our algorithms with properly tuned parameters work better than those in [11] in general.

5.2. Image denoising

Image denoising is to reconstruct a clean image from a noised observation. A common image restoration model is the well known total variation or Rudin-Osher-Fatemi [23] model:

$$\min_x |\nabla x| + \frac{\mu}{2}\|x - f\|^2, \tag{5.4}$$

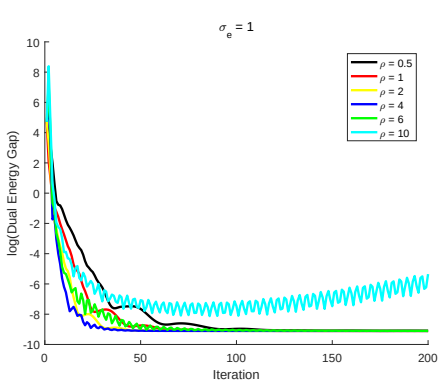
where x is the matrix representing the pixel value of the digital image, ∇x represents the discrete forward difference operator on x:

$$(\nabla x)_{i,j} = ((\nabla x)^1_{i,j}, (\nabla x)^2_{i,j})$$

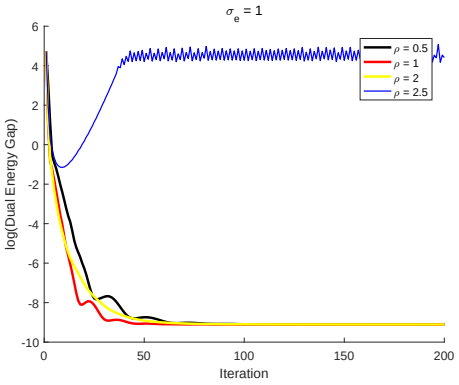
with

$$(\nabla x)^1_{i,j} = \begin{cases} x_{i+1,j} - x_{i,j}, & \text{if} \quad i < N, \\ 0, & \text{if} \quad i = N, \end{cases} \tag{5.5a}$$

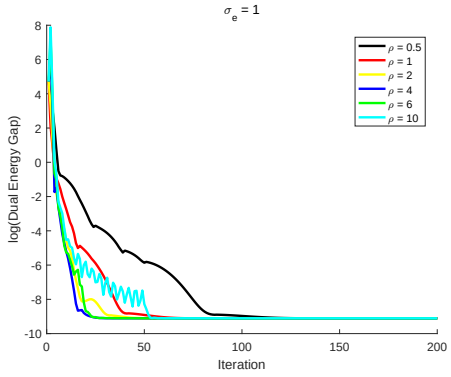
$$(\nabla x)^2_{i,j} = \begin{cases} x_{i,j+1} - x_{i,j}, & \text{if} \quad j < N, \\ 0, & \text{if} \quad j = N, \end{cases} \tag{5.5b}$$



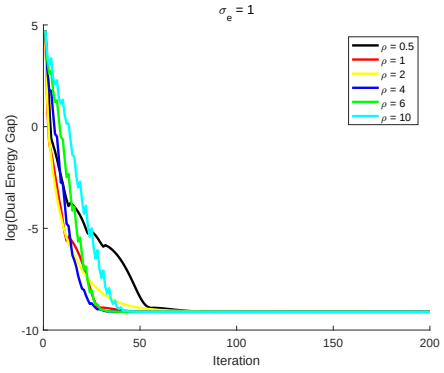
(a) Fadmm with different ρ



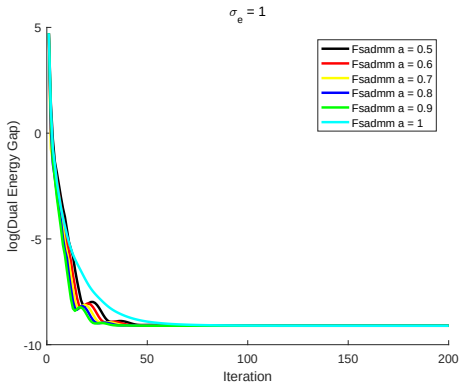
(b) Fsadmm with different ρ



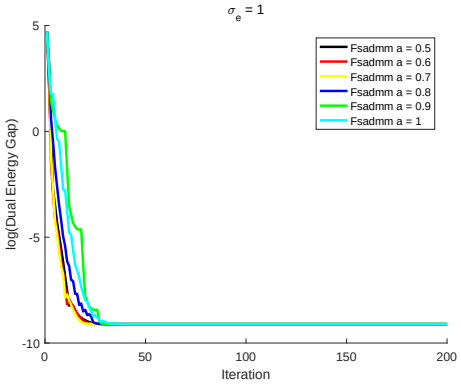
(c) Fadmm+ restart with different ρ



(d) Fsadmm + restart with different ρ

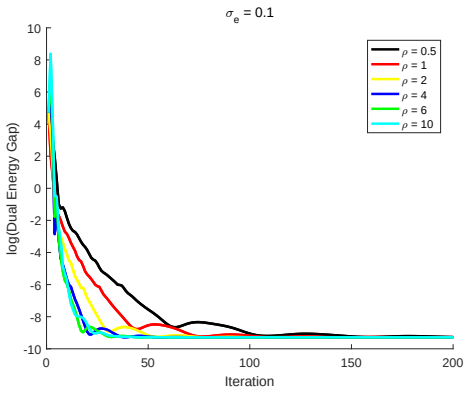


(e) Fsadmm with different a, $\rho = 2$

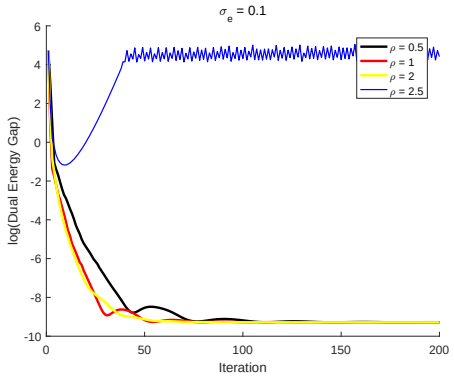


(f) Fsadmm + restart with different a, $\rho = 4$

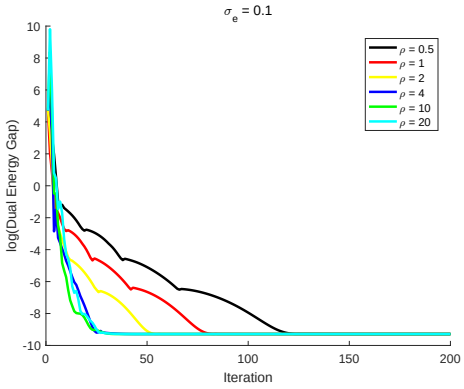
Figure 1: Different algorithms on well conditioned elastic net problem.



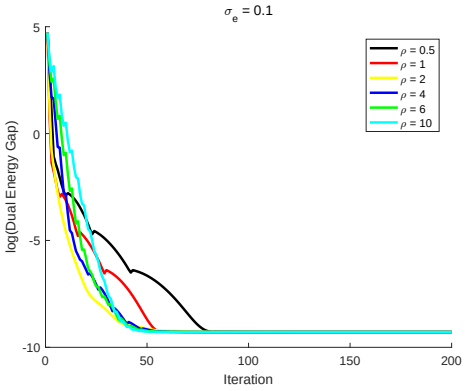
(a) Fadmm with different ρ



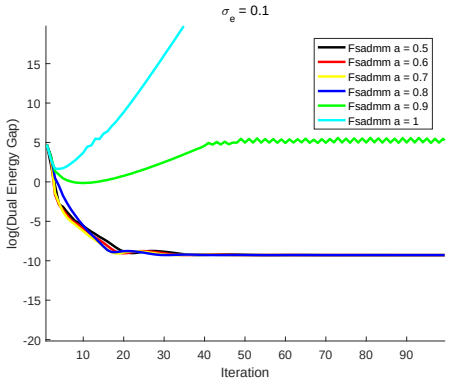
(b) Fsadmm with different ρ



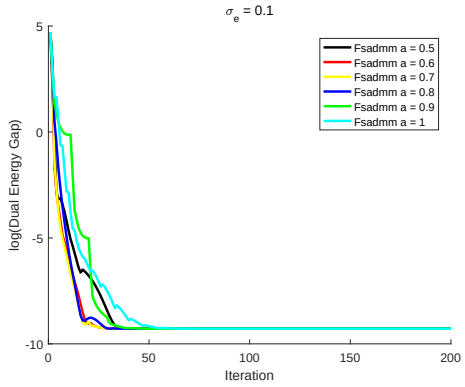
(c) Fadmm+ restart with different ρ



(d) Fsadmm + restart with different ρ

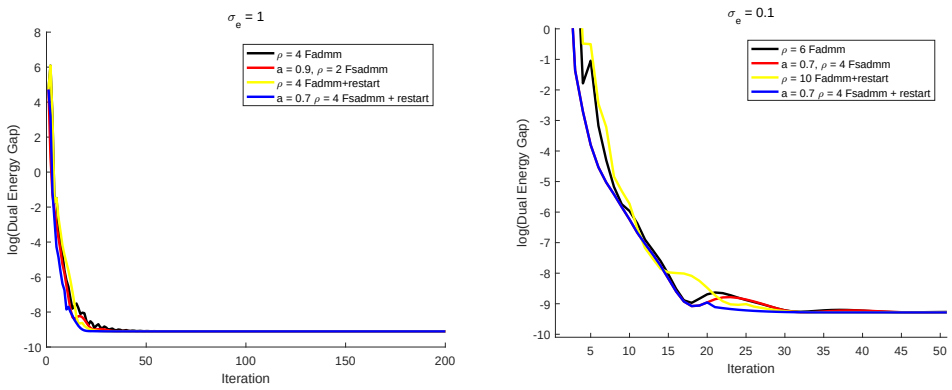


(e) Fsadmm with different a, $\rho = 4$



(f) Fsadmm + restart with different a, $\rho = 4$

Figure 2: Different algorithms on poor conditioned elastic net problem.



(a) well conditioned (b) poor conditioned

Figure 3: Algorithms with best parameters comparing on elastic net problem.

and $|\nabla x| = \sqrt{(\nabla x^1)^2 + (\nabla x^2)^2}$. Another definition is :

$$(\nabla x)_{i,j} = ((\nabla x)^1_{i,j}, (\nabla x)^2_{i,j})$$

with

$$(\nabla x)^1_{i,j} = \begin{cases} x_{i+1,j} - x_{i,j}, & \text{if} \quad i < N, \\ x_{1,j} - x_{N,j}, & \text{if} \quad i = N, \end{cases} \tag{5.6a}$$

$$(\nabla x)^2_{i,j} = \begin{cases} x_{i,j+1} - x_{i,j}, & \text{if} \quad j < N, \\ x_{i,1} - x_{i,N}, & \text{if} \quad j = N, \end{cases} \tag{5.6b}$$

and $|\nabla x| = |\nabla x^1| + |\nabla x^2|$. Here we adopt the latter one for simplicity.

Implement details. (5.4) can be reformulated as

$$\begin{cases} \min\limits_{x,y} |x| + \dfrac{\mu}{2}\|y - f\|^2 \\ \text{s.t.} \;\; x - \nabla y = 0. \end{cases} \tag{5.7}$$

In this case, x has two parts x_1 and x_2, for the separability of l_1 norm, we can divide the x-subproblem into two independent problem:

$$\min_{x_1} |x_1| - x^\top \lambda_1^k + \frac{\rho}{2}\|x_1 - (\nabla y^k)^1\|, \tag{5.8a}$$

$$\min_{x_2} |x_2| - x^\top \lambda_2^k + \frac{\rho}{2}\|x_2 - (\nabla y^k)^2\|. \tag{5.8b}$$

The Lagrange multiplier λ here is composed by λ_1 and λ_2. The above problems can be easily solved by a shrink operator which is same to the x-subproblem in the elastic net

(a) clean image

(b) noised image

(c) $\mu = 5$

(d) $\mu = 10$

(e) $\mu = 20$

Figure 4: The clean, blurred and reconstructed images of cameraman.

problem. The y-subproblem is

$$\min_y \mu\|y - f\|^2 + \rho\|x^k - \nabla y - \lambda^k/\rho\|^2. \tag{5.9}$$

The ∇ operator can be seen as a convolution operator under the above definition and after a Fourier Transform it become a diagonal matrix, then the problem (5.9) is easily solved through FFT.

From the definition $|\nabla x|$, it is not a strongly convex function of x, so we compare the performance of ADMM, sADMM and their accelerated variants. In this experiment, we test the algorithm on the cameraman image. we scale the pixel matrix from $0 \sim 255$ to $0 \sim 1$ and add a Gaussian noise with standard variance 0.01. For choosing μ properly is crucial for the quality of the recovered images, so a set of μ is tested. The original image, the noised image, the recovered image under different μ are shown in Fig. 4.

We choose the stopping criterion as

$$\frac{\|y^k - y^*\|^2}{\|y^*\|} \leq 10^{-3}, \tag{5.10}$$

where y^* represents the optimal solution of the problem. We compare the numbers of

Table 1: The numbers of iteration for different algorithms on image denoising.

Algorithms	$\mu = 5$	$\mu = 10$	$\mu = 20$
ADMM	124	83	27
sADMM $(a = 0.9)$	70	47	15
FADMM + restart	94	60	18
FsADMM + restart $(a = 0.7)$	86	55	16

iteration for different algorithms with best tuned parameter ρ and a. The results are listed in Table 1.

From Table 1, we see the Algorithm 4.1 works well in weakly convex condition. We see that the accelerated symmetric ADMM with restart has better convergence behavior than ADMM, accelerated ADMM with restart [11]. For the problem with bad condition, the restart procedure happens frequently and this is the reason that it takes a little more iterations than symmetric ADMM, however with better condition of the problem, Algorithm 4.1 will dramatically accelerate the original symmetric ADMM.

6. Conclusions

In this paper, we present a fast symmetric ADMM and prove its $\mathcal{O}(\frac{1}{k^2})$ convergence rate under strongly convex condition. For problems that do not meet the assumption, we present a modified algorithm with a restart rule which it is guaranteed to converge to an optimal solution. Numerical results shows that the accelerated symmetric ADMM and accelerated symmetric ADMM with restart work well on problem with different conditions and show a better potential than accelerated ADMM in [11].

Acknowledgements This research is partly supported by the National Natural Science Foundation of China (Grant No. 11671217) and Natural Science Foundation of Xinjiang (Grant No. 2017D01A14). The authors thank the reviewers for their valuable comments.

References

[1] AMIR BECK AND MARC TEBOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems,* SIAM J. Imag. Sci., 2(1) (2009), pp. 183–202.

[2] STEPHEN BOYD, NEAL PARIKH, ERIC CHU, BORJA PELEATO AND JONATHAN ECKSTEIN, *Distributed optimization and statistical learning via the alternating direction method of multipliers,* Foundations and Trends® in Machine Learning, 3(1) (2011), pp. 1–122.

[3] ANTONIN CHAMBOLLE AND THOMAS POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging,* J. Math. Imag. Vision, 40(1) (2011), pp. 120–145.

[4] CAIHUA CHEN, BINGSHENG HE, YINYU YE AND XIAOMING YUAN, *The direct extension of admm for multi-block convex minimization problems is not necessarily convergent,* Math. Program., 155(1-2) (2016), pp. 57–79.

[5] JIM DOUGLAS AND HENRY H. RACHFORD, *On the numerical solution of heat conduction problems in two and three space variables,* Transactions of the American Mathematical Society, 82(2) (1956), pp. 421–439.

[6] JONATHAN ECKSTEIN, *Parallel alternating direction multiplier decomposition of convex programs,* J. Optimization Theory Appl., 80(1) (1994), pp. 39–62.

[7] JONATHAN ECKSTEIN AND DIMITRI P. BERTSEKAS, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators,* Math. Program., 55(1-3) (1992), pp. 293–318.

[8] JONATHAN ECKSTEIN AND MICHAEL C. FERRIS, *Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control,* Informs J. Comput., 10(2) (1998), pp. 218–235.

[9] ROLAND GLOWINSKI AND A. MARROCO, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires, Revue française d'automatique, informatique, recherche opérationnelle, Analyse Numérique*, 9(2) (1975), pp. 41–76.

[10] DONALD GOLDFARB, SHIQIAN MA AND KATYA SCHEINBERG, *Fast alternating linearization methods for minimizing the sum of two convex functions,* Math. Program., 141(1-2) (2013), pp. 349–382.

[11] TOM GOLDSTEIN, BRENDAN O'DONOGHUE, SIMON SETZER AND RICHARD BARANIUK, *Fast alternating direction optimization methods,* SIAM J. Imag. Sci., 7(3) (2014), pp. 1588–1623.

[12] BINGSHENG HE, HAN LIU, ZHAORAN WANG AND XIAOMING YUAN, *A strictly contractive Peaceman-Rachford splitting method for convex programming,* SIAM J. Optimization, 24(3) (2014), pp. 1011–1040.

[13] BINGSHENG HE, HAI YANG AND SHENGLI WANG, *Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities,* J. Optimization Theory Appl., 106(2) (2000), pp. 337–356.

[14] BINGSHENG HE AND XIAOMING YUAN, *On the $\mathcal{O}(1/n)$ convergence rate of the Douglas-Rachford alternating direction method,* SIAM J. Numer. Anal., 50(2) (2012), pp. 700–709.

[15] BINGSHENG HE AND XIAOMING YUAN, *On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers,* Numer. Math., 130(3) (2015), pp. 567–577.

[16] MAGNUS R. HESTENES, *Multiplier and gradient methods,* J. Optimization Theory Appl., 4(5) (1969), pp. 303–320.

[17] MOJTABA KADKHODAIE, KONSTANTINA CHRISTAKOPOULOU, MAZIAR SANJABI AND ARINDAM BANERJEE, *Accelerated alternating direction method of multipliers,* in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 497–506, ACM, 2015.

[18] YURII NESTEROV, *A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$,* in Soviet Mathematics Doklady, Volume 27, pages 372–376, 1983.

[19] BRENDAN O'DONOGHUE AND EMMANUEL CANDES, *Adaptive restart for accelerated gradient schemes,* Found. Comput. Math., 15(3) (2015), pp. 715–732.

[20] YUYUAN OUYANG, YUNMEI CHEN, GUANGHUI LAN AND EDUARDO PASILIAO JR, *An accelerated linearized alternating direction method of multipliers,* SIAM J. Imag. Sci., 8(1) (2015), pp. 644–681.

[21] DONALD W PEACEMAN AND HENRY H. RACHFORD JR, *The numerical solution of parabolic and elliptic differential equations,* J. Society Industrial Appl. Math., 3(1) (1955), pp. 28–41.

[22] MICHAEL J. D. POWELL, *A method for nonlinear constraints in minimization problems,* Optimization, (1969), pp. 283–298.

[23] LEONID I. RUDIN, STANLEY OSHER AND EMAD FATEMI, *Nonlinear total variation based noise removal algorithms,* Physica D: Nonlinear Phenomena, 60(1-4) (1992), pp. 259–268.

[24] ANDRZEJ RUSZCZYŃSKI, *An augmented Lagrangian decomposition method for block diagonal linear programming problems,* Operations Res. Lett., 8(5) (1989), pp. 287–294.

[25] HANGRUI YUE, QINGZHI YANG, XIANGFENG WANG AND XIAOMING YUAN, *Implementing the alternating direction method of multipliers for big datasets: A case study of least absolute shrinkage and selection operator,* SIAM J. Sci. Comput., 40(5) (2018), pp. A3121–A3156.

[26] HUI ZOU AND TREVOR HASTIE, *Regularization and variable selection via the elastic net,* Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2) (2005), pp. 301–320.