

# An Extragradient-Based Alternating Direction Method for Convex Minimization

Tianyi Lin <sup>\*</sup>      Shiqian Ma <sup>†</sup>      Shuzhong Zhang <sup>‡</sup>

July 9, 2015

## Abstract

In this paper, we consider the problem of minimizing the sum of two convex functions subject to linear linking constraints. The classical alternating direction type methods usually assume that the two convex functions have relatively easy proximal mappings. However, many problems arising from statistics, image processing and other fields have the structure that while one of the two functions has easy proximal mapping, the other function is smoothly convex but does not have an easy proximal mapping. Therefore, the classical alternating direction methods cannot be applied. To deal with the difficulty, we propose in this paper an alternating direction method based on *extragredients*. Under the assumption that the smooth function has a Lipschitz continuous gradient, we prove that the proposed method returns an  $\epsilon$ -optimal solution within  $O(1/\epsilon)$  iterations. We apply the proposed method to solve a new statistical model called fused logistic regression. Our numerical experiments show that the proposed method performs very well when solving the test problems. We also test the performance of the proposed method through solving the lasso problem arising from statistics and compare the result with several existing efficient solvers for this problem; the results are very encouraging indeed.

**Keywords:** Alternating Direction Method; Extragradient; Iteration Complexity; Lasso; Fused Logistic Regression

**Mathematics Subject Classification 2010:** 90C25, 68Q25, 62J05

---

<sup>\*</sup>Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, N. T., Hong Kong. Email: linyt@se.cuhk.edu.hk.

<sup>†</sup>Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, N. T., Hong Kong. Email: sqma@se.cuhk.edu.hk. Research of this author was supported in part by a Direct Grant of The Chinese University of Hong Kong (Project ID: 4055016) and the Hong Kong Research Grants Council General Research Funds Early Career Scheme (Project ID: CUHK 439513).

<sup>‡</sup>Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, MN 55455. Email: zhangs@umn.edu. Research of this author was supported in part by the NSF Grant CMMI-1161242.

# 1 Introduction

In this paper, we consider solving the following convex optimization problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n, y \in \mathbb{R}^p} \quad & f(x) + g(y) \\ \text{s.t.} \quad & Ax + By = b \\ & x \in \mathcal{X}, y \in \mathcal{Y}, \end{aligned} \tag{1.1}$$

where  $f$  and  $g$  are convex functions,  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{m \times p}$ ,  $b \in \mathbb{R}^m$ ,  $\mathcal{X}$  and  $\mathcal{Y}$  are convex sets and the projections on them can be easily obtained. Problems in the form of (1.1) arise in different applications in practice and we will show some examples later. A recent very popular way to solve (1.1) is to apply the alternating direction method of multipliers (ADMM). A typical iteration of ADMM for solving (1.1) can be described as

$$\begin{cases} x^{k+1} & := \operatorname{argmin}_{x \in \mathcal{X}} \mathcal{L}_\gamma(x, y^k; \lambda^k) \\ y^{k+1} & := \operatorname{argmin}_{y \in \mathcal{Y}} \mathcal{L}_\gamma(x^{k+1}, y; \lambda^k) \\ \lambda^{k+1} & := \lambda^k - \gamma(Ax^{k+1} + By^{k+1} - b), \end{cases} \tag{1.2}$$

where the augmented Lagrangian function  $\mathcal{L}_\gamma(x, y; \lambda)$  for (1.1) is defined as

$$\mathcal{L}_\gamma(x, y; \lambda) := f(x) + g(y) - \langle \lambda, Ax + By - b \rangle + \frac{\gamma}{2} \|Ax + By - b\|^2, \tag{1.3}$$

where  $\lambda$  is the Lagrange multiplier associated with the linear constraint  $Ax + By = b$  and  $\gamma > 0$  is a penalty parameter. The ADMM is closely related to some operator splitting methods such as Douglas-Rachford operator splitting method [8] and Peaceman-Rachford operator splitting method [36] for finding the zero of the sum of two maximal monotone operators. In particular, it was shown by Gabay [14] that ADMM (1.2) is equivalent to applying the Douglas-Rachford operator splitting method to the dual problem of (1.1). The ADMM and operator splitting methods were then studied extensively in the literature and some generalized variants were proposed (see, e.g., [25, 12, 17, 9, 10]). The ADMM was revisited recently because it was found very efficient for solving many sparse and low-rank optimization problems, such as compressed sensing [50], compressive imaging [47, 18], robust PCA [43], sparse inverse covariance selection [53, 39], sparse PCA [27] and semidefinite programming [48] etc. Moreover, the iteration complexity of ADMM (1.2) was recently established by He and Yuan [21] and Monteiro and Svaiter [31], and some new analysis for iteration complexity for obtaining  $\epsilon$ -optimal solution measured by both objective error and constraint violation was given in [24] and [7]. The recent survey paper by Boyd *et al.* [4] listed many interesting applications of ADMM in statistical learning and distributed optimization.

Note that the efficiency of ADMM (1.2) actually depends on whether the two subproblems in (1.2) can be solved efficiently or not. This requires that the following two problems can be solved efficiently for given  $\tau > 0$ ,  $w, z \in \mathbb{R}^m$ :

$$x := \operatorname{argmin}_{x \in \mathcal{X}} f(x) + \frac{1}{2\tau} \|Ax - w\|^2 \tag{1.4}$$

and

$$y := \operatorname{argmin}_{y \in \mathcal{Y}} g(y) + \frac{1}{2\tau} \|By - z\|^2. \quad (1.5)$$

When  $\mathcal{X}$  and  $\mathcal{Y}$  are the whole spaces and  $A$  and  $B$  are identity matrices, (1.4) and (1.5) are known as the proximal mappings of functions  $f$  and  $g$ , respectively. Thus, in this case, ADMM (1.2) requires that the proximal mappings of  $f$  and  $g$  are easy to be obtained. In the cases that  $A$  and  $B$  are not identity matrices, there are results on linearized ADMM (see, e.g., [50, 49, 54]) which linearizes the quadratic penalty term in such a way that problems (1.4) and (1.5) still correspond to the proximal mappings of functions  $f$  and  $g$ . The global convergence of the linearized ADMM is guaranteed under certain conditions on a linearization step size parameter.

There are two interesting problems that are readily solved by ADMM (1.2) since the involved functions have easy proximal mappings. One problem is the so-called robust principal component pursuit (RPCP) problem:

$$\min_{X \in \mathbb{R}^{m \times n}, Y \in \mathbb{R}^{m \times n}} \|X\|_* + \rho \|Y\|_1, \text{ s.t., } X + Y = M, \quad (1.6)$$

where  $\rho > 0$  is a weighting parameter,  $M \in \mathbb{R}^{m \times n}$  is a given matrix,  $\|X\|_*$  is the nuclear norm of  $X$ , which is defined as the sum of singular values of  $X$ , and  $\|Y\|_1 := \sum_{i,j} |Y_{ij}|$  is the  $\ell_1$  norm of  $Y$ . Problem (1.6) was studied by Candès *et al.* [5] and Chandrasekaran *et al.* [6] as a convex relaxation of the robust PCA problem. Note that the two involved functions, the nuclear norm  $\|X\|_*$  and the  $\ell_1$  norm  $\|Y\|_1$ , have easy proximal mappings (see, e.g., [28] and [19]). The other problem is the so-called sparse inverse covariance selection, which is also known as the graphical lasso problem [52, 1, 13]. This problem, which estimates a sparse inverse covariance matrix from sample data, can be formulated as

$$\min_{X \in \mathbb{R}^{n \times n}} -\log \det(X) + \langle \hat{\Sigma}, X \rangle + \rho \|X\|_1, \quad (1.7)$$

where the first convex function  $-\log \det(X) + \langle \hat{\Sigma}, X \rangle$  is the negative log-likelihood function for given sample covariance matrix  $\hat{\Sigma}$ , and the second convex function  $\rho \|X\|_1$  is used to promote the sparsity of the resulting solution. Problem (1.7) is of the form of (1.1) because it can be rewritten equivalently as

$$\min_{X \in \mathbb{R}^{n \times n}, Y \in \mathbb{R}^{n \times n}} -\log \det(X) + \langle \hat{\Sigma}, X \rangle + \rho \|Y\|_1, \text{ s.t., } X - Y = 0. \quad (1.8)$$

Note that the involved function  $-\log \det(X)$  has an easy proximal mapping (see, e.g., [39] and [53]).

However, there are many problems arising from statistics, machine learning and image processing which do not have easy subproblems (1.4) and (1.5) even when  $A$  and  $B$  are identity matrices. One such example is the so-called sparse logistic regression problem. For given training set  $\{a_i, b_i\}_{i=1}^m$

where  $a_1, a_2, \dots, a_m$  are the  $m$  samples and  $b_1, \dots, b_m$  with  $b_i \in \{-1, +1\}, i = 1, \dots, m$  are the binary class labels. The likelihood function for these  $m$  samples is  $\prod_{i=1}^m \text{Prob}(b_i | a_i)$ , where

$$\text{Prob}(b | a) := \frac{1}{1 + \exp(-b(a^\top x + c))},$$

is the conditional probability of the label  $b$  condition on sample  $a$ , where  $x \in \mathbb{R}^n$  is the weight vector and  $c \in \mathbb{R}$  is the intercept, and  $a^\top x + c = 0$  defines a hyperplane in the feature space, on which  $\text{Prob}(b | a) = 0.5$ . Besides,  $\text{Prob}(b | a) > 0.5$  if  $a^\top x + c$  has the same sign as  $b$ , and  $\text{Prob}(b | a) < 0.5$  otherwise. The sparse logistic regression (see [26]) can be formulated as the following convex optimization problem

$$\min_{x,c} \ell(x, c) + \alpha \|x\|_1, \quad (1.9)$$

where  $\alpha > 0$  is a weighting parameter,  $\ell(x, c)$  denotes the average logistic loss function, which is defined as

$$\ell(x, c) := -\frac{1}{m} \prod_{i=1}^m \text{Prob}(b_i | a_i) = \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-b_i(a_i^\top x + c))),$$

and the  $\ell_1$  norm  $\|x\|_1$  is imposed to promote the sparsity of the weight vector  $x$ . If one wants to apply ADMM (1.2) to solve (1.9), one has to introduce a new variable  $y$  and rewrite (1.9) as

$$\begin{aligned} \min_{x,c,y} \quad & \ell(x, c) + \alpha \|y\|_1, \\ \text{s.t.} \quad & x - y = 0. \end{aligned} \quad (1.10)$$

When ADMM (1.2) is applied to solve (1.10), although the subproblem with respect to  $y$  is easily solvable (an  $\ell_1$  shrinkage operation), the subproblem with respect to  $(x, c)$  is difficult to solve because the proximal mapping of the logistic loss function  $\ell(x, c)$  is not easily computable.

Another example is the following fused logistic regression problem:

$$\min_{x,c} \ell(x, c) + \alpha \|x\|_1 + \beta \sum_{j=2}^n |x_j - x_{j-1}|, \quad (1.11)$$

or its constrained version:

$$\begin{aligned} \min_{x,c} \quad & \ell(x, c) + \alpha \|x\|_1 \\ \text{s.t.} \quad & \sum_{j=2}^n |x_j - x_{j-1}| \leq \xi, \end{aligned}$$

where  $\alpha, \beta$  and  $\xi$  are positive parameters. These problems cannot be solved by ADMM (1.2), again because of the difficulty in computing the proximal mapping of  $\ell(x, c)$ . We will discuss this example in more details in Section 4.

But is it really crucial to compute the proximal mapping exactly in the ADMM scheme? After all, ADMM can be viewed as an approximate dual gradient ascent method. As such, computing the proximal mapping exactly is in some sense redundant, since on the dual side the iterates are

updated based on the gradient ascent method. Without sacrificing the scale of approximation to optimality, an update on the primal side based on the gradient information (or at least part of it), by the principle of primal-dual symmetry, is entirely appropriate. Our subsequent analysis indeed confirms this belief. Moreover, there are inexact versions of augmented Lagrangian method and alternating direction method (see, e.g. [20, 11]), and our proposed method can be viewed as a variant of the inexact version of ADMM.

**Our contribution.** In this paper, we propose a new alternating direction method for solving (1.1). This new method requires only one of the functions in the objective to have an easy proximal mapping, and the other involved function is merely required to be smooth. Note that the aforementioned examples, namely sparse logistic regression (1.9) and fused logistic regression (1.11), are both of this type. In each iteration, the proposed method involves only computing the proximal mapping for one function and computing the gradient for the other function. Under the assumption that the smooth function has a Lipschitz continuous gradient, we prove that the proposed method finds an  $\epsilon$ -optimal solution to (1.1) within  $O(1/\epsilon)$  iterations.

The rest of this paper is organized as follows. In Section 2 we propose the extragradient-based alternating direction method for solving problem (1.1). The iteration complexity of the proposed method is analyzed in Section 3. In Section 4, we discuss the details of the fused logistic regression problem and how to use our proposed method to solve it. Numerical experiments are conducted in Section 5. Finally we draw some conclusions in Section 6.

## 2 An Alternating Direction Method Based on Extragradient

In this section, we consider solving (1.1) where  $f$  has an easy proximal mapping, while  $g$  is smooth but does not have an easy proximal mapping. Note that in this case, ADMM (1.2) cannot be applied to solve (1.1) as the solution for the second subproblem in (1.2) is not available.

Note that in ADMM (1.2), the updating formula for  $\lambda$  can be seen as a gradient step for  $\lambda$  with respect to the augmented Lagrangian function. Now because minimizing the augmented Lagrangian function with respect to  $y$  in (1.2) is not possible, we can also take a gradient step. However, our analysis shown later indicates that we have to take a gradient step for the Lagrangian function

$$\mathcal{L}(x, y; \lambda) := f(x) + g(y) - \langle \lambda, Ax + By - b \rangle \quad (2.1)$$

with respect to  $y$ . This leads to the following updating procedure for  $(x^k, y^k, \lambda^k)$ :

$$\begin{cases} x^{k+1} & := \operatorname{argmin}_{x \in \mathcal{X}} \mathcal{L}_\gamma(x, y^k; \lambda^k) + \frac{1}{2} \|x - x^k\|_H^2 \\ y^{k+1} & := [y^k - \gamma \nabla_y \mathcal{L}(x^{k+1}, y^k; \lambda^k)]_{\mathcal{Y}} \\ \lambda^{k+1} & := \lambda^k - \gamma (Ax^{k+1} + By^{k+1} - b), \end{cases} \quad (2.2)$$

where  $[y]_{\mathcal{Y}}$  denotes the projection of  $y$  onto  $\mathcal{Y}$ . Note that we have added a proximal term  $\frac{1}{2}\|x - x^k\|_H^2$  to the  $x$ -subproblem, where  $H$  is a pre-specified positive semidefinite matrix. In this paper, we propose to take an extra gradient step for both  $y$  and  $\lambda$ , which results in the following extragradient-based alternating direction method (EGADM) to solve Problem (1.1). Starting from any initial point  $x^0 \in \mathcal{X}$ ,  $y^0 \in \mathcal{Y}$  and  $\lambda^0 \in \mathbb{R}^m$ , a typical iteration of EGADM can be described as:

$$\begin{cases} x^{k+1} & := \operatorname{argmin}_{x \in \mathcal{X}} \mathcal{L}_\gamma(x, y^k; \lambda^k) + \frac{1}{2}\|x - x^k\|_H^2 \\ \bar{y}^{k+1} & := [y^k - \gamma \nabla_y \mathcal{L}(x^{k+1}, y^k; \lambda^k)]_{\mathcal{Y}} \\ \bar{\lambda}^{k+1} & := \lambda^k - \gamma(Ax^{k+1} + By^k - b) \\ y^{k+1} & := [y^k - \gamma \nabla_y \mathcal{L}(x^{k+1}, \bar{y}^{k+1}; \bar{\lambda}^{k+1})]_{\mathcal{Y}} \\ \lambda^{k+1} & := \lambda^k - \gamma(Ax^{k+1} + B\bar{y}^{k+1} - b). \end{cases} \quad (2.3)$$

Note that the first subproblem in (2.3) is to minimize the augmented Lagrangian function plus a proximal term  $\frac{1}{2}\|x - x^k\|_H^2$  with respect to  $x$ , i.e.,

$$x^{k+1} := \operatorname{argmin}_{x \in \mathcal{X}} f(x) - \langle \lambda^k, Ax + By^k - b \rangle + \frac{\gamma}{2}\|Ax + By^k - b\|^2 + \frac{1}{2}\|x - x^k\|_H^2. \quad (2.4)$$

In sparse and low-rank optimization problems, the proximal term  $\frac{1}{2}\|x - x^k\|_H^2$  is usually imposed to cancel the effect of matrix  $A$  in the quadratic penalty term. One typical choice of  $H$  is  $H = 0$  when  $A$  is identity, and  $H = \tau I - \gamma A^\top A$  when  $A$  is not identity, where  $\tau > \gamma \lambda_{\max}(A^\top A)$  and  $\lambda_{\max}(A^\top A)$  denotes the largest eigenvalue of  $A^\top A$ . We assume that (2.4) is relatively easy to solve. Basically, when  $A$  is an identity matrix, and  $f$  is a function arising from sparse optimization such as the  $\ell_1$  norm,  $\ell_2$  norm, nuclear norm and so on, then the subproblem (2.4) is usually easy to solve.

The EGADM (2.3) can be written equivalently in a more compact form. By defining

$$z := \begin{pmatrix} y \\ \lambda \end{pmatrix}, \bar{z} := \begin{pmatrix} \bar{y} \\ \bar{\lambda} \end{pmatrix}, F(x, z) = \begin{pmatrix} \nabla_y \mathcal{L}(x, y; \lambda) \\ -\nabla_\lambda \mathcal{L}(x, y; \lambda) \end{pmatrix}, \mathcal{Z} := \mathcal{Y} \times \mathbb{R}^m,$$

we can rewrite (2.3) equivalently as

$$\begin{cases} x^{k+1} & := \operatorname{argmin}_{x \in \mathcal{X}} \mathcal{L}_\gamma(x, y^k; \lambda^k) + \frac{1}{2}\|x - x^k\|_H^2 \\ \bar{z}^{k+1} & := [z^k - \gamma F(x^{k+1}, z^k)]_{\mathcal{Z}} \\ z^{k+1} & := [z^k - \gamma F(x^{k+1}, \bar{z}^{k+1})]_{\mathcal{Z}}. \end{cases} \quad (2.5)$$

That is, in each iteration of EGADM, one minimizes the augmented Lagrangian function plus a proximal term with respect to  $x$ , and then takes extragradient steps for both  $y$  and  $\lambda$  for the Lagrangian function.

The idea of extragradient is not new. In fact, the extragradient method as we know it was originally proposed by Korpelevich for variational inequalities and for solving saddle-point problems [22, 23]. Korpelevich proved the convergence of the extragradient method [22, 23]. For recent results on

convergence of extragradient type methods, we refer to [35] and the references therein. The iteration complexity of extragradient method was analyzed by Nemirovski in [33]. Recently, Monteiro and Svaiter [29, 30, 31] studied the iteration complexity results of the hybrid proximal extragradient method proposed by Solodov and Svaiter in [41] and its variants. More recently, Bonettini and Ruggiero studied a generalized extragradient method for total variation based image restoration problem [3].

### 3 Iteration Complexity

In this section, we analyze the iteration complexity of EGADM, i.e., (2.3). We show that under the assumption that the smooth function  $g$  has a Lipschitz continuous gradient, EGADM (2.3) finds an  $\epsilon$ -optimal solution to Problem (1.1) in terms of both objective error and constraint violation within  $O(1/\epsilon)$  iterations.

For the linearly constrained convex minimization (1.1), the most natural way to define the  $\epsilon$ -optimal solution is in terms of the objective error and constraint violation as follows (see also [24, 7]).

**Definition 3.1** *We call  $(\hat{x}, \hat{y}) \in \mathcal{X} \times \mathcal{Y}$  an  $\epsilon$ -optimal solution to Problem (1.1) in terms of objective error and constraint violation, if the following holds,*

$$|f(\hat{x}) + g(\hat{y}) - f(x^*) - g(y^*)| = O(\epsilon), \quad \text{and} \quad \|A\hat{x} + B\hat{y} - b\| = O(\epsilon) \quad (3.1)$$

where  $(x^*, y^*) \in \mathcal{X}^* \times \mathcal{Y}^*$  is any optimal solution to (1.1).

Throughout this paper, we assume that the primal-dual optimal solution sets  $\mathcal{X}^* \times \mathcal{Y}^* \times \Lambda^*$  of (1.1) is non-empty. Note that  $(x^*, y^*; \lambda^*)$  is an optimal solution to (1.1) if and only if  $x^* \in \mathcal{X}$ ,  $y^* \in \mathcal{Y}$ , and

$$\begin{cases} \langle x - x^*, \partial f(x^*) - A^\top \lambda^* \rangle \geq 0, & \forall x \in \mathcal{X}, \\ \langle y - y^*, \partial g(y^*) - B^\top \lambda^* \rangle \geq 0, & \forall y \in \mathcal{Y}, \\ Ax^* + By^* = b, \end{cases} \quad (3.2)$$

where  $\partial f(x)$  and  $\partial g(y)$  denote subgradients of  $f$  and  $g$ .

Now we are ready to analyze the iteration complexity of EGADM (2.5), or equivalently, (2.3), for an  $\epsilon$ -optimal solution in the sense of Definition 3.1. We will prove the following lemma first.

**Lemma 3.1** *The sequence  $\{x^{k+1}, z^k, \bar{z}^k\}$  generated by (2.5) satisfies the following inequality:*

$$\begin{aligned} & \langle \gamma F(x^{k+1}, \bar{z}^{k+1}), \bar{z}^{k+1} - z^{k+1} \rangle - \frac{1}{2} \|z^k - z^{k+1}\|^2 \\ \leq & \gamma^2 \|F(x^{k+1}, \bar{z}^{k+1}) - F(x^{k+1}, z^k)\|^2 - \frac{1}{2} \|\bar{z}^{k+1} - z^k\|^2 - \frac{1}{2} \|\bar{z}^{k+1} - z^{k+1}\|^2. \end{aligned} \quad (3.3)$$

*Proof.* Note that the optimality conditions of the two subproblems for  $z$  in (2.5) are given by

$$\langle z^k - \gamma F(x^{k+1}, z^k) - \bar{z}^{k+1}, z - \bar{z}^{k+1} \rangle \leq 0, \quad \forall z \in \mathcal{Z}, \quad (3.4)$$

and

$$\langle z^k - \gamma F(x^{k+1}, \bar{z}^{k+1}) - z^{k+1}, z - z^{k+1} \rangle \leq 0, \quad \forall z \in \mathcal{Z}. \quad (3.5)$$

Letting  $z = z^{k+1}$  in (3.4) and  $z = \bar{z}^{k+1}$  in (3.5), and then summing the two resulting inequalities, we get

$$\|z^{k+1} - \bar{z}^{k+1}\|^2 \leq \gamma \langle F(x^{k+1}, z^k) - F(x^{k+1}, \bar{z}^{k+1}), z^{k+1} - \bar{z}^{k+1} \rangle, \quad (3.6)$$

which implies

$$\|z^{k+1} - \bar{z}^{k+1}\| \leq \gamma \|F(x^{k+1}, z^k) - F(x^{k+1}, \bar{z}^{k+1})\|. \quad (3.7)$$

Now we are able to prove (3.3). We have,

$$\begin{aligned} & \langle \gamma F(x^{k+1}, \bar{z}^{k+1}), \bar{z}^{k+1} - z^{k+1} \rangle - \frac{1}{2} \|z^k - z^{k+1}\|^2 \\ = & \gamma \langle F(x^{k+1}, \bar{z}^{k+1}) - F(x^{k+1}, z^k), \bar{z}^{k+1} - z^{k+1} \rangle \\ & + \gamma \langle F(x^{k+1}, z^k), \bar{z}^{k+1} - z^{k+1} \rangle - \frac{1}{2} \|z^k - z^{k+1}\|^2 \\ \leq & \gamma \langle F(x^{k+1}, \bar{z}^{k+1}) - F(x^{k+1}, z^k), \bar{z}^{k+1} - z^{k+1} \rangle \\ & + \langle z^k - \bar{z}^{k+1}, \bar{z}^{k+1} - z^{k+1} \rangle - \frac{1}{2} \|z^k - z^{k+1}\|^2 \\ = & \gamma \langle F(x^{k+1}, \bar{z}^{k+1}) - F(x^{k+1}, z^k), \bar{z}^{k+1} - z^{k+1} \rangle \\ & - \frac{1}{2} \|z^k\|^2 + \langle z^k, \bar{z}^{k+1} \rangle + \langle \bar{z}^{k+1}, z^{k+1} - \bar{z}^{k+1} \rangle - \frac{1}{2} \|z^{k+1}\|^2 \\ \leq & \gamma \|F(x^{k+1}, \bar{z}^{k+1}) - F(x^{k+1}, z^k)\| \cdot \|\bar{z}^{k+1} - z^{k+1}\| \\ & + \left(-\frac{1}{2} \|z^k\|^2 + \langle z^k, \bar{z}^{k+1} \rangle - \frac{1}{2} \|\bar{z}^{k+1}\|^2\right) + \left(-\frac{1}{2} \|\bar{z}^{k+1}\|^2 + \langle \bar{z}^{k+1}, z^{k+1} \rangle - \frac{1}{2} \|z^{k+1}\|^2\right) \\ \leq & \gamma^2 \|F(x^{k+1}, \bar{z}^{k+1}) - F(x^{k+1}, z^k)\|^2 - \frac{1}{2} \|\bar{z}^{k+1} - z^k\|^2 - \frac{1}{2} \|\bar{z}^{k+1} - z^{k+1}\|^2, \end{aligned} \quad (3.8)$$

where the first inequality is obtained by letting  $z = z^{k+1}$  in (3.4) and the last inequality follows from (3.7). This completes the proof.  $\square$

We next prove the following lemma.

**Lemma 3.2** *Assume that  $\nabla g(y)$  is Lipschitz continuous with Lipschitz constant  $L_g$ , i.e.,*

$$\|\nabla g(y_1) - \nabla g(y_2)\| \leq L_g \|y_1 - y_2\|, \quad \forall y_1, y_2 \in \mathcal{Y}. \quad (3.9)$$

By letting  $\gamma \leq 1/(2\hat{L})$ , where  $\hat{L} := (\max\{2L_g^2 + \lambda_{\max}(B^\top B), 2\lambda_{\max}(B^\top B)\})^{\frac{1}{2}}$ , the following inequality holds,

$$\langle \gamma F(x^{k+1}, \bar{z}^{k+1}), \bar{z}^{k+1} - z^{k+1} \rangle - \frac{1}{2} \|z^k - z^{k+1}\|^2 \leq 0. \quad (3.10)$$



*Proof.* For any  $z_1 \in \mathcal{Z}$  and  $z_2 \in \mathcal{Z}$ , we have,

$$\begin{aligned}
& \|F(x^{k+1}, z_1) - F(x^{k+1}, z_2)\|^2 \\
&= \left\| \begin{pmatrix} (\nabla g(y_1) - B^\top \lambda_1) - (\nabla g(y_2) - B^\top \lambda_2) \\ (Ax^{k+1} + By_1 - b) - (Ax^{k+1} + By_2 - b) \end{pmatrix} \right\|^2 \\
&= \|(\nabla g(y_1) - \nabla g(y_2)) - B^\top(\lambda_1 - \lambda_2)\|^2 + \|B(y_1 - y_2)\|^2 \\
&\leq 2\|\nabla g(y_1) - \nabla g(y_2)\|^2 + 2\|B^\top(\lambda_1 - \lambda_2)\|^2 + \|B(y_1 - y_2)\|^2 \\
&\leq 2L_g^2\|y_1 - y_2\|^2 + 2\lambda_{\max}(B^\top B)\|\lambda_1 - \lambda_2\|^2 + \lambda_{\max}(B^\top B)\|y_1 - y_2\|^2 \\
&\leq \max\{2L_g^2 + \lambda_{\max}(B^\top B), 2\lambda_{\max}(B^\top B)\} \left\| \begin{pmatrix} y_1 - y_2 \\ \lambda_1 - \lambda_2 \end{pmatrix} \right\|^2 \\
&= \hat{L}^2\|z_1 - z_2\|^2,
\end{aligned}$$

where the second inequality is due to (3.9) and the last equality is from the definition of  $\hat{L}$ . Thus, we know that  $F(x^{k+1}, z)$  is Lipschitz continuous with Lipschitz constant  $\hat{L}$ . Since  $\gamma \leq 1/(2\hat{L})$ , we have the following inequality,

$$\begin{aligned}
& \gamma^2\|F(x^{k+1}, \bar{z}^{k+1}) - F(x^{k+1}, z^k)\|^2 - \frac{1}{2}\|\bar{z}^{k+1} - z^k\|^2 - \frac{1}{2}\|\bar{z}^{k+1} - z^{k+1}\|^2 \\
&\leq \gamma^2\|F(x^{k+1}, \bar{z}^{k+1}) - F(x^{k+1}, z^k)\|^2 - \frac{1}{2}\|\bar{z}^{k+1} - z^k\|^2 \\
&\leq (\gamma^2\hat{L}^2 - \frac{1}{2})\|\bar{z}^{k+1} - z^k\|^2 \\
&\leq 0,
\end{aligned}$$

which combining with (3.3) yields (3.10).  $\square$

We further prove the following lemma.

**Lemma 3.3** *Under the same assumptions as in Lemma 3.2, the following holds:*

$$\frac{1}{2}\|z - z^{k+1}\|^2 - \frac{1}{2}\|z - z^k\|^2 \leq \langle \gamma F(x^{k+1}, \bar{z}^{k+1}), z - \bar{z}^{k+1} \rangle. \quad (3.11)$$

*Proof.* Adding

$$\langle \gamma F(x^{k+1}, \bar{z}^{k+1}), z - z^{k+1} \rangle - \frac{1}{2}\|z^{k+1} - z^k\|^2$$

to both sides of (3.5), we get,

$$\langle z^k - z^{k+1}, z - z^{k+1} \rangle - \frac{1}{2}\|z^{k+1} - z^k\|^2 \leq \langle \gamma F(x^{k+1}, \bar{z}^{k+1}), z - z^{k+1} \rangle - \frac{1}{2}\|z^{k+1} - z^k\|^2. \quad (3.12)$$

Notice that the left hand side of (3.12) is equal to  $\frac{1}{2}\|z - z^{k+1}\|^2 - \frac{1}{2}\|z - z^k\|^2$ . Thus we have,

$$\begin{aligned}
& \frac{1}{2}\|z - z^{k+1}\|^2 - \frac{1}{2}\|z - z^k\|^2 \\
&\leq \langle \gamma F(x^{k+1}, \bar{z}^{k+1}), z - z^{k+1} \rangle - \frac{1}{2}\|z^{k+1} - z^k\|^2 \\
&= \langle \gamma F(x^{k+1}, \bar{z}^{k+1}), z - \bar{z}^{k+1} \rangle + \langle \gamma F(x^{k+1}, \bar{z}^{k+1}), \bar{z}^{k+1} - z^{k+1} \rangle - \frac{1}{2}\|z^{k+1} - z^k\|^2 \\
&\leq \langle \gamma F(x^{k+1}, \bar{z}^{k+1}), z - \bar{z}^{k+1} \rangle,
\end{aligned} \quad (3.13)$$

where the last inequality is due to (3.10).  $\square$

We now give the  $O(1/\epsilon)$  iteration complexity of (2.3) (or equivalently, (2.5)) for an  $\epsilon$ -optimal solution to Problem (1.1).

**Theorem 3.4** *Consider Algorithm EGADM (2.3), and its sequence of iterates. For any integer  $N > 0$ , define*

$$\tilde{x}^N := \frac{1}{N} \sum_{k=0}^N x^{k+1}, \quad \tilde{y}^N := \frac{1}{N} \sum_{k=0}^N \bar{y}^{k+1}, \quad \tilde{\lambda}^N := \frac{1}{N} \sum_{k=0}^N \bar{\lambda}^{k+1}.$$

Assume that  $\nabla g(y)$  is Lipschitz continuous with Lipschitz constant  $L_g$ , and we choose  $\gamma \leq 1/(2\hat{L})$ , where  $\hat{L} := (\max\{2L_g^2 + \lambda_{\max}(B^\top B), 2\lambda_{\max}(B^\top B)\})^{\frac{1}{2}}$ . Moreover, we choose  $H := 0$  if  $A$  is an identity matrix, and  $H := \tau I - \gamma A^\top A$  when  $A$  is not identity, where  $\tau > \gamma \lambda_{\max}(A^\top A)$ . For any optimal solution  $(x^*, y^*, \lambda^*) \in \mathcal{X}^* \times \mathcal{Y}^* \times \Lambda^*$  to (1.1), it holds that

$$|f(\tilde{x}^N) + g(\tilde{y}^N) - f(x^*) - g(y^*)| = O(1/N), \quad \text{and} \quad \|A\tilde{x}^N + B\tilde{y}^N - b\| = O(1/N). \quad (3.14)$$

Note that (3.14) implies that when  $N = O(1/\epsilon)$ ,  $\{\tilde{x}^N, \tilde{y}^N, \tilde{\lambda}^N\}$  is an  $\epsilon$ -optimal solution to Problem (1.1) in the sense of Definition 3.1; i.e., the iteration complexity of (2.3) (or equivalently, (2.5)) for an  $\epsilon$ -optimal solution to Problem (1.1) is  $O(1/\epsilon)$  in terms of both objective error and constraint violation.

*Proof.* The optimality conditions of the subproblem for  $x$  in (2.3) are given by

$$\langle \theta^{k+1} - A^\top \lambda^k + \gamma A^\top (Ax^{k+1} + By^k - b) + H(x^{k+1} - x^k), x - x^{k+1} \rangle \geq 0, \quad \forall x \in \mathcal{X}, \quad (3.15)$$

where  $\theta^{k+1} \in \partial f(x^{k+1})$  denotes a subgradient of  $f$  at point  $x^{k+1}$ . By using the updating formula for  $\bar{\lambda}^{k+1}$  in (2.3), i.e.,

$$\bar{\lambda}^{k+1} := \lambda^k + \gamma \nabla_{\lambda} \mathcal{L}(x^{k+1}, y^k; \lambda^k) = \lambda^k - \gamma (Ax^{k+1} + By^k - b),$$

we obtain,

$$\langle \theta^{k+1} - A^\top \bar{\lambda}^{k+1} + H(x^{k+1} - x^k), x - x^{k+1} \rangle \geq 0, \quad \forall x \in \mathcal{X}. \quad (3.16)$$

Combining (3.11) and (3.16), we have,

$$\begin{aligned} & \left\langle \begin{pmatrix} x - x^{k+1} \\ y - \bar{y}^{k+1} \\ \lambda - \bar{\lambda}^{k+1} \end{pmatrix}, \begin{pmatrix} \theta^{k+1} - A^\top \bar{\lambda}^{k+1} \\ \nabla g(\bar{y}^{k+1}) - B^\top \bar{\lambda}^{k+1} \\ Ax^{k+1} + B\bar{y}^{k+1} - b \end{pmatrix} \right\rangle \\ & \geq \frac{1}{2\gamma} (\|z - z^{k+1}\|^2 - \|z - z^k\|^2) + \langle H(x^k - x^{k+1}), x - x^{k+1} \rangle. \end{aligned} \quad (3.17)$$

Note that by using the convexity of functions  $f$  and  $g$  and letting  $x = x^*$  and  $y = y^*$ , (3.17) implies that the following holds for any  $\lambda \in \mathbb{R}^m$ :

$$\begin{aligned}
& f(x^*) + g(y^*) - f(x^{k+1}) - g(\bar{y}^{k+1}) + \begin{pmatrix} x^* - x^{k+1} \\ y^* - \bar{y}^{k+1} \\ \lambda - \bar{\lambda}^{k+1} \end{pmatrix}^\top \begin{pmatrix} -A^\top \bar{\lambda}^{k+1} \\ -B^\top \bar{\lambda}^{k+1} \\ Ax^{k+1} + B\bar{y}^{k+1} - b \end{pmatrix} \\
& + \left( \frac{1}{2} \|x^* - x^k\|_H^2 + \frac{1}{2\gamma} \|y^* - y^k\|^2 + \frac{1}{2\gamma} \|\lambda - \lambda^k\|^2 \right) \\
& - \left( \frac{1}{2} \|x^* - x^{k+1}\|_H^2 + \frac{1}{2\gamma} \|y^* - y^{k+1}\|^2 + \frac{1}{2\gamma} \|\lambda - \lambda^{k+1}\|^2 \right) \\
& \geq 0,
\end{aligned}$$

which further yields that

$$\begin{aligned}
& f(x^*) + g(y^*) - f(\tilde{x}^N) - g(\tilde{y}^N) + \lambda^\top (A\tilde{x}^N + B\tilde{y}^N - b) \\
= & f(x^*) + g(y^*) - f(\tilde{x}^N) - g(\tilde{y}^N) + \begin{pmatrix} x^* - \tilde{x}^N \\ y^* - \tilde{y}^N \\ \lambda - \tilde{\lambda}^N \end{pmatrix}^\top \begin{pmatrix} -A^\top \tilde{\lambda}^N \\ -B^\top \tilde{\lambda}^N \\ A\tilde{x}^N + B\tilde{y}^N - b \end{pmatrix} \\
\geq & \frac{1}{N+1} \sum_{k=0}^N \left[ f(x^*) + g(y^*) - f(x^{k+1}) - g(\bar{y}^{k+1}) + \begin{pmatrix} x^* - x^{k+1} \\ y^* - \bar{y}^{k+1} \\ \lambda - \bar{\lambda}^{k+1} \end{pmatrix}^\top \begin{pmatrix} -A^\top \bar{\lambda}^{k+1} \\ -B^\top \bar{\lambda}^{k+1} \\ Ax^{k+1} + B\bar{y}^{k+1} - b \end{pmatrix} \right] \\
\geq & \frac{1}{N+1} \sum_{k=0}^N \left[ \left( \frac{1}{2} \|x^* - x^{k+1}\|_H^2 + \frac{1}{2\gamma} \|y^* - y^{k+1}\|^2 + \frac{1}{2\gamma} \|\lambda - \lambda^{k+1}\|^2 \right) \right. \\
& \left. - \left( \frac{1}{2} \|x^* - x^k\|_H^2 + \frac{1}{2\gamma} \|y^* - y^k\|^2 + \frac{1}{2\gamma} \|\lambda - \lambda^k\|^2 \right) \right] \\
\geq & -\frac{1}{2\gamma(N+1)} \|\lambda - \lambda^0\|^2 - \frac{1}{2\gamma(N+1)} \|y^* - y^0\|^2 - \frac{1}{2(N+1)} \|x^* - x^0\|_H^2, \tag{3.18}
\end{aligned}$$

where the first inequality is due to the convexity of  $f$  and  $g$ . Note that the optimality condition (3.2) and convexity of  $f$  and  $g$  imply the following inequality

$$0 \geq f(x^*) + g(y^*) - f(\tilde{x}^N) - g(\tilde{y}^N) + \langle \lambda^*, A\tilde{x}^N + B\tilde{y}^N - b \rangle. \tag{3.19}$$

Now, define  $\rho := \|\lambda^*\| + 1$ . By using Cauchy-Schwarz inequality in (3.19), we obtain

$$0 \leq f(\tilde{x}^N) + g(\tilde{y}^N) - f(x^*) - g(y^*) + \rho \|A\tilde{x}^N + B\tilde{y}^N - b\|. \tag{3.20}$$

By setting  $\lambda = -\rho(A\tilde{x}^N + B\tilde{y}^N - b)/\|A\tilde{x}^N + B\tilde{y}^N - b\|$  in (3.18), and noting that  $\|\lambda\| = \rho$ , we obtain

$$\begin{aligned}
& f(\tilde{x}^N) + g(\tilde{y}^N) - f(x^*) - g(y^*) + \rho \|A\tilde{x}^N + B\tilde{y}^N - b\| \\
\leq & \frac{\rho^2 + \|\lambda^0\|^2}{\gamma(N+1)} + \frac{1}{2\gamma(N+1)} \|y^* - y^0\|^2 + \frac{1}{2(N+1)} \|x^* - x^0\|_H^2. \tag{3.21}
\end{aligned}$$

We now define the function

$$v(\xi) = \min\{f(x) + g(y) \mid Ax + By - b = \xi, x \in \mathcal{X}, y \in \mathcal{Y}\}.$$

It is easy to verify that  $v$  is convex,  $v(0) = f(x^*) + g(y^*)$ , and  $\lambda^* \in \partial v(0)$ . Therefore, from the convexity of  $v$ , it holds that

$$v(\xi) \geq v(0) + \langle \lambda^*, \xi \rangle \geq f(x^*) + g(y^*) - \|\lambda^*\| \|\xi\|. \quad (3.22)$$

Let  $\bar{\xi} = A\tilde{x}^N + B\tilde{y}^N - b$ , we have  $f(\tilde{x}^N) + g(\tilde{y}^N) \geq v(\bar{\xi})$ . Therefore, combining (3.20), (3.21) and (3.22), we get

$$\begin{aligned} -\|\lambda^*\| \|\bar{\xi}\| &\leq f(\tilde{x}^N) + g(\tilde{y}^N) - f(x^*) - g(y^*) \\ &\leq \frac{\rho^2 + \|\lambda^0\|^2}{\gamma(N+1)} + \frac{1}{2\gamma(N+1)} \|y^* - y^0\|^2 + \frac{1}{2(N+1)} \|x^* - x^0\|_H^2 - \rho \|\bar{\xi}\|, \end{aligned}$$

which by defining

$$C := \frac{\rho^2 + \|\lambda^0\|^2}{\gamma} + \frac{\|y^* - y^0\|^2}{2\gamma} + \frac{1}{2} \|x^* - x^0\|_H^2$$

yields,

$$\|A\tilde{x}^N + B\tilde{y}^N - b\| = \|\bar{\xi}\| \leq \frac{C}{N+1}, \quad (3.23)$$

and

$$-\frac{\|\lambda^*\| C}{N+1} \leq f(\tilde{x}^N) + g(\tilde{y}^N) - f(x^*) - g(y^*) \leq \frac{C}{N+1}. \quad (3.24)$$

Combining (3.23) and (3.24) completes the proof.  $\square$

**Remark 1** We note that there is another line of research on studying dual smoothing methods for solving (1.1). This approach is studied in [32, 46]. The dual smoothing method in [32] smoothes the Lagrangian dual function using Nesterov's smoothing technique in [34], and then applies the accelerated gradient method [34] to solve the smoothed dual problem. The accelerated gradient method can return an  $\epsilon$ -optimal solution to (1.1) in  $O(1/\epsilon)$  iterations. It should be noted that the smoothing technique used in [32] requires the feasible sets  $\mathcal{X}$  and  $\mathcal{Y}$  to be bounded, while this assumption is not needed in our EGADM method. On the other hand, the dual smoothing method in [32] does not assume any smoothness assumptions on  $f$  and  $g$ , while our EGADM requires  $g$  to be differentiable. In [46], the authors propose to smooth the Lagrangian dual function using self-concordant barrier functions, and then apply a path-following gradient method to minimize the smoothed dual function. It is noted that this method also requires the feasible sets  $\mathcal{X}$  and  $\mathcal{Y}$  to be bounded, and the  $O(1/\epsilon)$  complexity result is only measured by the dual objective value error.

## 4 Fused Logistic Regression

In this section we show how to apply Algorithm (2.3) to solve the fused logistic regression problem, which is a convex problem. To introduce our fused logistic regression model, we need to introduce fused lasso problem and logistic regression first. The sparse linear regression problem, known as Lasso [44], was introduced to find sparse regression coefficients so that the resulting model is more interpretable. The original Lasso model solves the following problem:

$$\min \frac{1}{2} \|Ax - b\|^2, \text{ s.t. } \|x\|_1 \leq s, \quad (4.1)$$

where  $A = [a_1, \dots, a_m]^\top \in \mathbb{R}^{m \times n}$  gives the predictor variables,  $b = [b_1, \dots, b_m]^\top \in \mathbb{R}^m$  gives the responses and the constraint  $\|x\|_1 \leq s$  is imposed to promote the sparsity of the regression coefficients  $x$ . The Lasso solution  $x$  gives more interpretability to the regression model since the sparse solution  $x$  ensures that only a few features contribute to the prediction.

Fused lasso was introduced by Tibshirani *et al.* in [45] to model the situation in which there is certain natural ordering in the features. Fused lasso adds a term to impose the sparsity of  $x$  in the gradient space to model natural ordering in the features. The fused lasso problem can be formulated as

$$\min \frac{1}{2} \|Ax - b\|^2 + \alpha \|x\|_1 + \beta \sum_{j=2}^n |x_j - x_{j-1}|. \quad (4.2)$$

Because (4.2) can be transformed equivalently to a quadratic programming problem, Tibshirani *et al.* proposed to solve (4.2) using a two-phase active set algorithm SQOPT of Gill *et al.* [16]. However, transforming (4.2) to a quadratic programming problem will increase the size of the problem significantly, thus SQOPT can only solve (4.2) with small or medium sizes. Ye and Xie [51] proposed to solve (4.2) using split Bregman algorithm, which can be shown to be equivalent to an alternating direction method of multipliers. Note that (4.2) can be rewritten equivalently as

$$\begin{aligned} \min \quad & \frac{1}{2} \|Ax - b\|^2 + \alpha \|w\|_1 + \beta \|y\|_1 \\ \text{s.t.} \quad & w = x \\ & y = Lx, \end{aligned} \quad (4.3)$$

where  $L$  is an  $(n - 1) \times n$  dimensional matrix with all ones on the diagonal and negative ones on the super-diagonal and zeros elsewhere. The ADMM can be applied to solve (4.3) with  $x$  being one block variable and  $(w, y)$  being the other block (see [51] for more details).

As in [26], the sparse logistic regression problem (1.9) can be formulated as

$$\min_{x, c} \ell(x, c), \text{ s.t. } \|x\|_1 \leq s. \quad (4.4)$$

It is now very meaningful to consider the fused logistic regression problem when there is certain natural ordering in the features. This leads to the following optimization problem:

$$\min_{x \in \mathbb{R}^n, c \in \mathbb{R}} \ell(x, c) + \alpha \|x\|_1 + \beta \sum_{j=2}^n |x_j - x_{j-1}|. \quad (4.5)$$

Problem (4.5) can be rewritten equivalently as

$$\begin{aligned} \min_{x \in \mathbb{R}^n, w \in \mathbb{R}^{n-1}, y \in \mathbb{R}^n, c \in \mathbb{R}} \quad & \alpha \|x\|_1 + \beta \|w\|_1 + \ell(y, c) \\ \text{s.t.} \quad & x = y \\ & w = Ly. \end{aligned} \quad (4.6)$$

If we apply the ADMM to solve (4.6), we will end up with the following iterates:

$$\begin{cases} (x^{k+1}, w^{k+1}) & := \operatorname{argmin}_{x, w} \mathcal{L}_\gamma(x, w, y^k, c^k; \lambda_1^k, \lambda_2^k) \\ (y^{k+1}, c^{k+1}) & := \operatorname{argmin}_{y, c} \mathcal{L}_\gamma(x^{k+1}, w^{k+1}, y, c; \lambda_1^k, \lambda_2^k) \\ \lambda_1^{k+1} & := \lambda_1^k - \gamma(x^{k+1} - y^{k+1}) \\ \lambda_2^{k+1} & := \lambda_2^k - \gamma(w^{k+1} - Ly^{k+1}), \end{cases} \quad (4.7)$$

where the augmented Lagrangian function  $\mathcal{L}_\gamma(x, w, y; \lambda_1, \lambda_2)$  is defined as

$$\begin{aligned} \mathcal{L}_\gamma(x, w, y, c; \lambda_1, \lambda_2) := & \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-b_i(a_i^\top y + c))) \\ & + \alpha \|x\|_1 + \beta \|w\|_1 - \langle \lambda_1, x - y \rangle - \langle \lambda_2, w - Ly \rangle + \frac{\gamma}{2} \|x - y\|^2 + \frac{\gamma}{2} \|w - Ly\|^2. \end{aligned}$$

However, note that although the subproblem for  $(x, w)$  is still easy, the subproblem for  $y$  is no longer easy because of the logistic loss function  $\ell(y, c)$ . But, since  $\ell(y, c)$  is differentiable with respect to  $(y, c)$ , we can apply our EGADM to solve (4.6). Noting that the subproblem for  $(x, w)$  corresponds to two  $\ell_1$  shrinkage operations. Moreover, the gradients of  $\ell(y, c)$  with respect to  $y$  and  $c$  are easily obtainable as

$$\nabla_y \ell(y, c) = -\frac{1}{m} \hat{A}^\top (1 - d), \quad \nabla_c \ell(y, c) = -\frac{1}{m} b^\top (1 - d), \quad d = 1./\alpha (1 + \exp(-\hat{A}y - cb)), \quad (4.8)$$

where  $\hat{A} = [b_1 a_1, b_2 a_2, \dots, b_m a_m]^\top$  and  $1./\alpha$  denotes the component-wise division.

We are now ready to describe the details of EGADM for solving (4.6) in Algorithm 1, in which the  $\ell_1$  shrinkage operation is defined as:

$$\text{Shrink}(z, \tau) := \operatorname{sign}(z) \circ \max\{|z| - \tau, 0\}.$$

## 5 Numerical Experiments

In this section, we test the performance of our EGADM for solving the fused logistic regression problem (4.5) and lasso problem. Our codes were written in MATLAB. All numerical experiments were run in MATLAB 7.12.0 on a laptop with Intel Core I5 2.5 GHz CPU and 4GB of RAM.

---

**Algorithm 1** Extragradient-based ADM for the Fused Logistic Regression
 

---

 Initialization:  $\hat{A} = [b_1 a_1, b_2 a_2, \dots, b_m a_m]^\top$ 
**for**  $k = 0, 1, \dots$  **do**

$$x^{k+1} := \text{Shrink}(y^k + \lambda_1^k / \gamma, \alpha / \gamma)$$

$$w^{k+1} := \text{Shrink}(Ly^k + \lambda_2^k / \gamma, \beta / \gamma)$$

$$d^k := 1. / (1 + \exp(-\hat{A}y^k - b \circ c^k)), \nabla_y \ell(y^k, c^k) := -\frac{1}{m} \hat{A}^\top (1 - d^k), \nabla_c \ell(y^k, c^k) := -\frac{1}{m} b^\top (1 - d^k)$$

$$\bar{y}^{k+1} := y^k - \gamma(\nabla_y \ell(y^k, c^k) + \lambda_1^k + L^\top \lambda_2^k)$$

$$\bar{c}^{k+1} := c^k - \gamma \nabla_c \ell(y^k, c^k)$$

$$\bar{\lambda}_1^{k+1} := \lambda_1^k - \gamma(x^{k+1} - y^k), \bar{\lambda}_2^{k+1} := \lambda_2^k - \gamma(w^{k+1} - Ly^{k+1})$$

$$\bar{d}^{k+1} := 1. / (1 + \exp(-\hat{A}\bar{y}^{k+1} - b \circ \bar{c}^{k+1}))$$

$$\nabla_y \ell(\bar{y}^{k+1}, \bar{c}^{k+1}) := -\frac{1}{m} \hat{A}^\top (1 - \bar{d}^{k+1}), \nabla_c \ell(\bar{y}^{k+1}, \bar{c}^{k+1}) := -\frac{1}{m} b^\top (1 - \bar{d}^{k+1})$$

$$y^{k+1} := y^k - \gamma(\nabla_y \ell(\bar{y}^{k+1}, \bar{c}^{k+1}) + \bar{\lambda}_1^{k+1} + L^\top \bar{\lambda}_2^{k+1})$$

$$c^{k+1} := c^k - \gamma \nabla_c \ell(\bar{y}^{k+1}, \bar{c}^{k+1})$$

$$\lambda_1^{k+1} := \lambda_1^k - \gamma(x^{k+1} - \bar{y}^{k+1}), \lambda_2^{k+1} := \lambda_2^k - \gamma(w^{k+1} - Ly^{k+1})$$

**end for**


---

## 5.1 Numerical Results for Fused Logistic Regression

In this subsection, we report the results of our EGADM (Algorithm 1) for solving the fused logistic regression problem (4.5).

First, we used a very simple example to show that when the features have natural ordering, the fused logistic regression model (4.5) is much preferable than the sparse logistic regression model (4.4). This simple example was created in the following manner. We created the regression coefficient  $\hat{x} \in \mathbb{R}^n$  for  $n = 1000$  as

$$\hat{x}_j = \begin{cases} r_1, & j = 1, 2, \dots, 100 \\ r_2, & j = 201, 202, \dots, 300 \\ r_3, & j = 401, 402, \dots, 500 \\ r_4, & j = 601, 602, \dots, 700 \\ 0, & \text{else,} \end{cases} \quad (5.1)$$

where scalers  $r_1, r_2, r_3, r_4$  were created randomly uniform in  $(0, 20)$ . An example plot of  $\hat{x}$  is shown in the left part of Figure 1. The entries of matrix  $A \in \mathbb{R}^{m \times n}$  with  $m = 500$  and  $n = 1000$  were drawn from standard normal distribution  $\mathcal{N}(0, 1)$ . Vector  $b \in \mathbb{R}^m$  was then created as the signs of  $A\hat{x} + ce$ , where  $c$  is a random number in  $(0, 1)$  and  $e$  is the  $m$ -dimensional vector of all ones. We then applied our extragradient-based ADM (Algorithm 1) for solving the fused logistic regression problem (4.5) and compared the result with the sparse logistic regression problem (4.4). The code for solving (4.4), which is called Lassplore and proposed by Liu *et al.* in [26], was downloaded from <http://www.public.asu.edu/~jye02/Software/lassplore/>. Default settings of Lassplore were used.

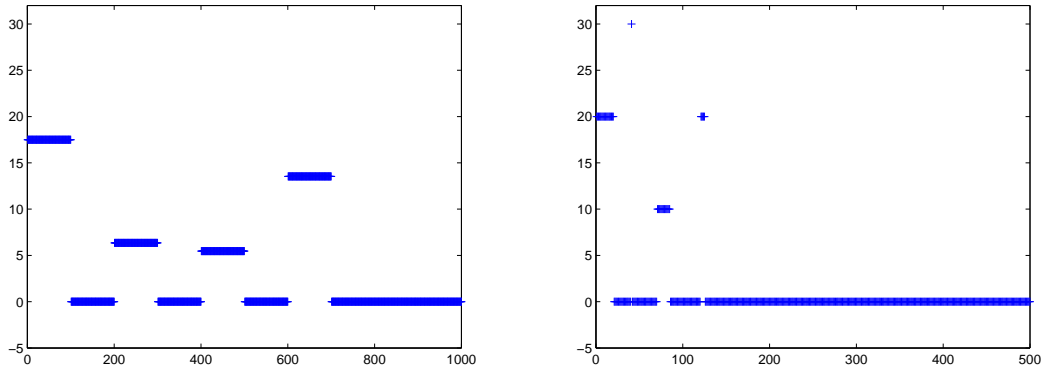


Figure 1: Left: The regression coefficient given in (5.1); Right: The regression coefficient given in (5.2).

We chose  $\alpha = 5 \times 10^{-4}$  and  $\beta = 5 \times 10^{-2}$  in (4.5). The regression result by EGADM (Algorithm 1) is plotted in Figure 2 (a). We tested different choices of  $s = 1, 5, 10$  in (4.4) and the results are plotted in Figure 2 (b), (c) and (d), respectively. From Figure 2 we see that, the fused logistic regression model (4.5) can preserve the natural ordering very well. The sparse logistic regression model (4.4) gives very sparse solution when  $s$  is small, and gives less sparse solution when  $s$  is large, but none of the choices of  $s = 1, 5, 10$  gives a solution that preserves the natural ordering.

To further show the capability of our EGADM for solving the fused logistic regression model (4.5), especially for large-scale problems, we conducted the following tests. First, we created the regression coefficient  $\hat{x} \in \mathbb{R}^n$  for  $n \geq 100$  as

$$\hat{x}_j = \begin{cases} 20, & j = 1, 2, \dots, 20, \\ 30, & j = 41, \\ 10, & j = 71, \dots, 85, \\ 20, & j = 121, \dots, 125, \\ 0, & \text{else.} \end{cases} \quad (5.2)$$

Note that a similar test example was used in [51] for the fused lasso problem. An example plot of  $\hat{x}$  of size  $n = 500$  is shown in the right part of Figure 1. We then created matrix  $A$  and vector  $b$  in the same way mentioned above. We applied our EGADM to solve the fused logistic regression model (4.5) with the above mentioned inputs  $A$  and  $b$ . We report the iteration number, CPU time, sparsity of  $x$  (denoted by  $\|x\|_0$ ) and sparsity of the fused term  $Lx$  (denoted by  $\|Lx\|_0$ ) in Table 1. From Table 1 we see that our EGADM can solve the fused logistic regression problem (4.5) efficiently. It solved instances with size up to  $m = 2000$ ,  $n = 20000$  in just a few seconds.



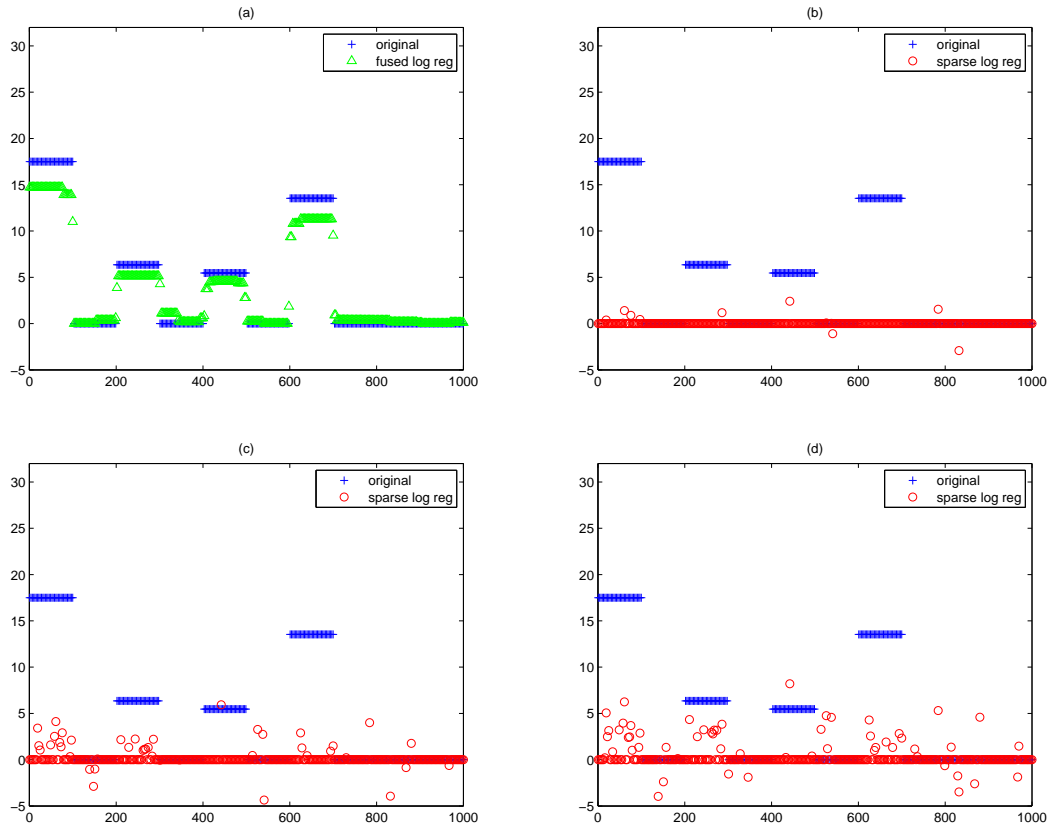


Figure 2: (a): The regression result by the fused logistic regression model (4.5); (b), (c), (d): The regression result by the sparse logistic regression model (4.4) with  $s = 1, 5, 10$ , respectively.

Table 1: Numerical Results for Fused Logistic Regression

$m$	$n$	iter	cpu	$\ x\ _0$	$\ Lx\ _0$
100	500	104	0.0	37	40
100	1000	112	0.1	41	46
100	2000	105	0.2	71	88
1000	2000	69	0.6	25	21
1000	5000	79	1.6	25	26
1000	10000	53	2.0	25	25
2000	5000	94	3.5	40	36
2000	10000	238	17.1	40	6
2000	20000	84	11.5	40	42

## 5.2 Numerical Results for Lasso Problem

To understand how our new method compares to other well established methods when all are applicable, in this subsection, we experiment our EGADM in comparison with ISTA [2] and ADMM for solving the following unconstrained version of the lasso problem (4.1):

$$\min \tau \|x\|_1 + \frac{1}{2} \|Ax - b\|^2, \quad (5.3)$$

where  $\tau > 0$  is a given weighting parameter. This problem can be naturally solved by ISTA [2]. A typical iteration of ISTA for solving (5.3) can be described as:

$$x^{k+1} := \operatorname{argmin}_x \gamma \tau \|x\|_1 + \frac{1}{2} \|x - (x^k - \gamma A^\top (Ax^k - b))\|^2, \quad (5.4)$$

where  $\gamma > 0$  is the step size of the gradient step. By applying a variable-splitting technique, (5.3) can be equivalently written as

$$\min \tau \|x\|_1 + \frac{1}{2} \|Ay - b\|^2, \text{ s.t.}, x - y = 0, \quad (5.5)$$

which can be solved by both EGADM and ADMM. A typical iteration of EGADM for solving (5.5) is

$$\begin{cases} x^{k+1} & := \operatorname{argmin}_x \tau \|x\|_1 - \langle \lambda^k, x - y^k \rangle + \frac{\gamma}{2} \|x - y^k\|^2 \\ \bar{y}^{k+1} & := y^k - \gamma (A^\top (Ay^k - b) + \lambda^k) \\ \bar{\lambda}^{k+1} & := \lambda^k - \gamma (x^{k+1} - y^k) \\ y^{k+1} & := y^k - \gamma (A^\top (A\bar{y}^{k+1} - b) + \bar{\lambda}^{k+1}) \\ \lambda^{k+1} & := \lambda^k - \gamma (x^{k+1} - \bar{y}^{k+1}), \end{cases} \quad (5.6)$$

and a typical iteration of ADMM for solving (5.5) is

$$\begin{cases} x^{k+1} & := \operatorname{argmin}_x \tau \|x\|_1 - \langle \lambda^k, x - y^k \rangle + \frac{\gamma}{2} \|x - y^k\|^2 \\ y^{k+1} & := \operatorname{argmin}_y \frac{1}{2} \|Ay - b\|^2 - \langle \lambda^k, x^{k+1} - y \rangle + \frac{\gamma}{2} \|x^{k+1} - y\|^2 \\ \lambda^{k+1} & := \lambda^k - \gamma(x^{k+1} - y^{k+1}). \end{cases} \quad (5.7)$$

The main computational efforts in these three algorithms are as follows. In each iteration of ISTA, an  $\ell_1$  shrinkage operation and two matrix vector multiplications are needed; in each iteration of EGADM, an  $\ell_1$  shrinkage operation and four matrix vector multiplications are needed; while in each iteration of ADMM, an  $\ell_1$  shrinkage operation and solving one linear system are required. Note that the lasso problem is suitable for ADMM because the  $y$ -subproblem is still easy to solve. We use this problem as an example to illustrate the relation among ISTA, EGADM and ADMM. In fact, even if the  $y$ -subproblem is not easy to solve, an inexact version of ADMM can still be applied. For instance, instead of solving the  $y$ -subproblem in (5.7) directly by solving a linear system, we can run gradient descent method for  $M$  iterations to get an approximate solution to it. Therefore, in our experiments for solving the lasso problem (5.3), we implemented the following comparison to compare the performance of ISTA, EGADM and ADMM. In our numerical experiments, we always set  $\tau = 0.1$  for simplicity. The instances for different  $(m, n)$  were created randomly in the following way. We first generated matrix  $A \in \mathbb{R}^{m \times n}$  randomly according to normal distribution. We then normalized  $A$  such that the largest singular value of  $A$  is 1. This normalization was implemented so that the step size  $\gamma$  can be selected easily. A sparse  $x$  was then created such that the number of nonzero components was equal to  $n/10$ , and their positions were selected uniformly randomly. We then set  $b = Ax$ . We first ran ISTA with  $\gamma = 1$  for 100 iterations, and recorded the resulting objective function value of (5.3) (denoted by  $f_I$ ). We then ran EGADM and ADMM with different  $\gamma$  until the objective function value was smaller than  $f_I$ , or the maximum number of iteration (set as 1000) was achieved. Moreover, we also ran the inexact version ADMM described above for different  $M$  to see the comparison result. The results are reported in Tables 2 and 3. In particular, ADMM-5 and ADMM-10 indicate that we respectively ran the gradient descent method for 5 and 10 iterations to get an approximate solution to the  $y$ -subproblem in (5.7);  $mvm$  denotes the total number of matrix vector multiplications. In Tables 2 and 3, we reported the results for different  $\gamma$ . Note that we used the same  $\gamma$  for ISTA, EGADM and ADMM. Moreover, the same  $\gamma$  was also used as the step size for the gradient descent method for solving the  $y$ -subproblems in ADMM-5 and ADMM-10.

From Tables 2 and 3 we have the following observations. When  $\gamma = 1.0$  and  $m < n$ , the number of iterations of EGADM, ADMM, ADMM-5 and ADMM-10 are similar with ISTA, while ADMM is more costly because it needs to solve an  $n \times n$  linear system in each iteration. When  $\gamma = 1.0$  and  $m > n$ , ADMM is better than other solvers, because now the linear system is in a relatively small size. Moreover, EGADM also performs well when  $m > n$ . It needs less iteration number than ISTA (except one instance). However, ADMM-5 and ADMM-10 do not perform well, because the step

size  $\gamma$  is too large. When  $\gamma$  is modest, i.e.,  $\gamma = 0.8$  and  $\gamma = 0.5$ , ADMM, ADMM-5 and ADMM-10 usually perform better than ISTA in terms of number of iterations. EGADM is comparable to ISTA in these two cases in the sense that EGADM needs more iterations than ISTA when  $m < n$  and less iterations when  $m > n$ . When  $\gamma = 0.1$ , ADMM-5, ADMM-10 and EGADM are worse than ISTA, but we emphasize here that ISTA always uses a large step size  $\gamma = 1$ .

Based on these comparison results, we may conclude as follows. First, ADMM is usually better than ISTA and EGADM when the two subproblems can be easily solved. Second, ADMM is not very sensitive to the selection of the penalty parameter  $\gamma$ , while ISTA and EGADM depends on it because these two methods are more like gradient method and need to choose step size properly. In the case that one subproblem is not easy to solve in ADMM, running gradient descent method for several iterations to get an approximate solution to the subproblem usually works. However, choosing the step size  $\gamma$  is very crucial to the performance. Furthermore, we emphasize again that ISTA only works for problems without linear constraints, and ADMM works when both subproblems can be solved easily. In contrast, EGADM provides something in the middle: it works when linear constraints are present and one subproblem is not easy to solve.

**Remark 2** *We remark here that as shown in many references, the current state-of-the-art algorithm for solving the lasso problem (5.3), is the randomized coordinate descent method (see, e.g, [37, 38, 40]). In our numerical comparison conducted in this subsection, we only compared our EGADM with ISTA and ADMM. We did not compare with the randomized coordinate descent method because EGADM, ISTA and ADMM are all batch type methods and we want to focus the comparison among the efficient batch methods for lasso problem. Moreover, stochastic ADMM as studied in [42] can solve the dual problem of (1.1). It is suitable for solving problems where  $f$  is the empirical loss function in machine learning or data fitting problems. For example, it can be applied to solve the lasso problem (5.3) or the so-called graph lasso problem. However, the method in [42] assumes that the proximal mappings of both  $f$  and  $g$  are easily computable. As a result, it is not suitable for solving the fused logistic regression problem (4.5).*

**Remark 3** *Since our proposed EGADM method is a variant of ADMM, it is capable of solving problems at least as large as that can be handled by ADMM. Moreover, the main computational effort in each iteration of EGADM is computing the proximal mapping of function  $f$  and computing the gradient of function  $g$ . EGADM is efficient as long as these computations can be done relatively easily. From Tables 1, 2 and 3 we see that EGADM is efficient for at least medium-sized problems. For even larger problems, we may implement some parallel or randomized versions of EGADM which will be left as future research topics.*

## 6 Concluding Remarks

In this paper, we proposed a new alternating direction method based on extragradient for solving convex minimization problems with the objective function being the sum of two convex functions. The proposed method applies to the situation where only one of the involved functions has easy proximal mapping, while the other function is only known to be smooth. Under the assumption that the smooth function has a Lipschitz continuous gradient, we proved that the proposed method finds an  $\epsilon$ -optimal solution within  $O(1/\epsilon)$  iterations. We used the lasso problem to illustrate the performance of the proposed method and compared its performance with some existing solvers for this problem. We also proposed a new statistical model, namely fused logistic regression, that can preserve the natural ordering of the features in logistic regression. Preliminary numerical results showed that this new model is preferable than the sparse logistic regression model when there exists natural ordering in the features. The numerical results also showed that our extragradient-based ADM can solve large-scale fused logistic regression model efficiently.

It is noted that we only considered problems with two block variables in this paper. If there are  $N$ -block variables with  $N \geq 3$ , the proposed extragradient-based ADM is still applicable. In particular, if the first  $N - 1$  functions have easy proximal mappings and the last one does not but is smooth, we can apply the multi-block ADMM to solve this problem, but replacing the minimization for the augmented Lagrangian function with respect to the last block variable by an extragradient step. However, the convergence properties of this algorithm are currently not known, and this will be a topic for the future research. One simplification of EGADM is *GADM*; that is, replacing the extra-gradient steps by a single gradient step. The method was proposed in an earlier version of this paper, whose numerical performance was found to be comparable with EGADM; however, the convergence status of *GADM* was unknown to us. Very recently, [15] resolved the issue and proved that the iteration complexity of *GADM* remains  $O(1/\epsilon)$ .

## Acknowledgements

We thank Lingzhou Xue and Hui Zou for fruitful discussions on logistic regression and fused lasso. We are also grateful to two anonymous referees for their constructive comments that have helped improve the presentation of this paper greatly.

## References

- [1] O. Banerjee, L. El Ghaoui, and A. d’Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian for binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.
- [2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.
- [3] S. Bonettini and V. Ruggiero. An alternating extragradient method for total variation based image restoration from Poisson data. *Inverse Problems*, 27:095001, 2011.
- [4] 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.
- [5] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of ACM*, 58(3):1–37, 2011.
- [6] V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.
- [7] D. Davis and W. Yin. Convergence rate analysis of several splitting schemes. *UCLA CAM Report 14-51*, 2014.
- [8] J. Douglas and H. H. Rachford. On the numerical solution of the heat conduction problem in 2 and 3 space variables. *Transactions of the American Mathematical Society*, 82:421–439, 1956.
- [9] J. Eckstein. *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis, Massachusetts Institute of Technology, 1989.
- [10] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318, 1992.
- [11] J. Eckstein and P. J. S. Silva. A practical relative error criterion for augmented lagrangians. *Mathematical Programming*, 141:319–348, 2013.
- [12] M. Fortin and R. Glowinski. *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems*. North-Holland Pub. Co., 1983.
- [13] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

- [14] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Solution of Boundary Value Problems*. North-Holland, Amsterdam, 1983.
- [15] X. Gao, B. Jiang, and S. Zhang. On the information-adaptive variants of the ADMM: an iteration complexity perspective. *Optimization Online*, 11/4633, 2014.
- [16] P. E. Gill, W. Murray, and M. A. Saunders. Users guide for SQOPT 5.3: a Fortran package for large-scale linear and quadratic programming. Technical report, Technical Report NA 97-4. University of California, San Diego., 1997.
- [17] R. Glowinski and P. Le Tallec. *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM, Philadelphia, Pennsylvania, 1989.
- [18] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.*, 2:323–343, 2009.
- [19] E. T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for  $\ell_1$ -minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.
- [20] B. He, L. Liao, D. Han, and H. Yang. A new inexact alternating direction method for monotone variational inequalities. *Mathematical Programming*, 92:103–118, 2002.
- [21] B. He and X. Yuan. On the  $\mathcal{O}(1/n)$  convergence rate of douglas-rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50:700–709, 2012.
- [22] G. Korpelevich. The extragradient method for finding saddle points and other problems. *Ekonomika i Matematicheskie Metody*, 12:747–756, 1976. (in Russian; English translation in Matekon).
- [23] G. Korpelevich. Extrapolation gradient methods and relation to modified lagrangeans. *Ekonomika i Matematicheskie Metody*, 19:694–703, 1983. (in Russian; English translation in Matekon).
- [24] T. Lin, S. Ma, and S. Zhang. On the sublinear convergence rate of multi-block ADMM. *Journal of the Operations Research Society of China*, 2015.
- [25] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16:964–979, 1979.
- [26] J. Liu, J. Chen, and J. Ye. Large-scale sparse logistic regression. In *SIGKDD*, 2009.
- [27] S. Ma. Alternating direction method of multipliers for sparse principal component analysis. *Journal of the Operations Research Society of China*, 1(2):253–274, 2013.

- [28] S. Ma, D. Goldfarb, and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming Series A*, 128:321–353, 2011.
- [29] R. D. C. Monteiro and B. F. Svaiter. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM Journal on Optimization*, 20:2755–2787, 2010.
- [30] R. D. C. Monteiro and B. F. Svaiter. Complexity of variants of Tseng’s modified F-B splitting and Korpelevich’s methods for hemi-variational inequalities with applications to saddle point and convex optimization problems. *SIAM Journal on Optimization*, 21:1688–1720, 2011.
- [31] R. D. C. Monteiro and B. F. Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *SIAM Journal on Optimization*, 23:475–507, 2013.
- [32] I. Necoara and J. Suykens. Application of a smoothing technique to decomposition in convex optimization. *IEEE Trans. Automat. Contr.*, 53(11):2674–2679, 2008.
- [33] A. Nemirovski. Prox-method with rate of convergence  $O(1/t)$  for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2005.
- [34] Y. E. Nesterov. Smooth minimization for non-smooth functions. *Math. Program. Ser. A*, 103:127–152, 2005.
- [35] M. A. Noor. New extragradient-type methods for general variational inequalities. *Journal of Mathematical Analysis and Applications*, 277(2):379–394, 2003.
- [36] D. H. Peaceman and H. H. Rachford. The numerical solution of parabolic elliptic differential equations. *SIAM Journal on Applied Mathematics*, 3:28–41, 1955.
- [37] P. Richtarik and M. Takac. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(2):1–38, 2014.
- [38] P. Richtarik and M. Takac. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 2015.
- [39] K. Scheinberg, S. Ma, and D. Goldfarb. Sparse inverse covariance selection via alternating linearization methods. In *NIPS*, 2010.
- [40] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 2015.



- [41] M. V. Solodov and B. F. Svaiter. A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator. *Set-Valued Anal.*, 7:323–345, 1999.
- [42] T. Suzuki. Stochastic dual coordinate ascent with alternating direction multiplier method. In *ICML*, 2014.
- [43] M. Tao and X. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM J. Optim.*, 21:57–81, 2011.
- [44] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal Royal Statistical Society B*, 58:267–288, 1996.
- [45] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(1):91–108, 2005.
- [46] Q. Tran Dinh, I. Necoara, and M. Diehl. Path-following gradient-based decomposition algorithms for separable convex optimization. *Journal of Global Optimization*, 59:59–80, 2014.
- [47] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.
- [48] Z. Wen, D. Goldfarb, and W. Yin. Alternating direction augmented Lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, 2:203–230, 2010.
- [49] J. Yang and X. Yuan. Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of Computation*, 82(281):301–329, 2013.
- [50] J. Yang and Y. Zhang. Alternating direction algorithms for  $\ell_1$  problems in compressive sensing. *SIAM Journal on Scientific Computing*, 33(1):250–278, 2011.
- [51] G. Ye and X. Xie. Split Bregman method for large scale fused Lasso. *Computational Statistics and Data Analysis*, 55(4):1552–1569, 2011.
- [52] M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.
- [53] X. Yuan. Alternating direction methods for sparse covariance selection. *Journal of Scientific Computing*, 51:261–273, 2012.
- [54] X. Zhang, M. Burger, X. Bresson, and S. Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Science*, 3:253–276, 2010.

Table 2: Numerical Results for Lasso problem

	ISTA		ADMM		ADMM-5			ADMM-10			EGADM		
$(m, n)$	iter	cpu	iter	cpu	iter	mvm	cpu	iter	mvm	cpu	iter	mvm	cpu
$\gamma = 1.0$													
(100,1000)	100	0.1	101	5.2	102	510	0.1	1000	10000	0.7	102	408	0.0
(100,2000)	100	0.3	101	30.7	103	515	0.1	604	6040	0.9	102	408	0.1
(100,5000)	100	0.8	102	435.9	105	525	0.6	220	2200	2.2	102	408	0.2
(100,8000)	100	1.3	102	1642.8	106	530	3.7	321	3210	5.9	102	408	0.5
(1000,100)	100	0.1	53	0.0	1000	5000	0.3	1000	10000	0.6	108	432	0.0
(1000,200)	100	0.3	88	0.1	1000	5000	0.5	1000	10000	1.0	165	660	0.0
(2000,200)	100	0.6	55	0.1	1000	5000	2.5	1000	10000	4.3	99	396	0.1
(5000,100)	100	0.7	35	0.0	1000	5000	4.6	1000	10000	8.2	40	160	0.1
(5000,200)	100	1.5	43	0.1	1000	5000	10.4	1000	10000	19.6	66	264	0.3
(8000,100)	100	1.3	45	0.1	1000	5000	8.5	1000	10000	16.2	46	184	0.2
(8000,200)	100	2.3	46	0.2	1000	5000	16.4	1000	10000	31.2	62	248	0.5
$\gamma = 0.8$													
(100,1000)	100	0.1	81	3.9	82	410	0.0	81	810	0.1	127	508	0.0
(100,2000)	100	0.3	81	23.1	82	410	0.1	81	810	0.1	127	508	0.1
(100,5000)	100	0.8	81	320.3	82	410	0.5	82	820	0.7	127	508	0.3
(100,8000)	100	1.2	81	1295.8	82	410	0.8	82	820	1.4	127	508	0.6
(1000,100)	100	0.1	42	0.0	44	220	0.0	42	420	0.0	57	228	0.0
(1000,200)	100	0.3	62	0.1	53	265	0.0	55	550	0.1	79	316	0.0
(2000,200)	100	0.6	1000	1.4	44	220	0.1	1000	10000	4.5	65	260	0.1
(5000,100)	100	0.7	31	0.0	30	150	0.1	30	300	0.2	36	144	0.1
(5000,200)	100	1.4	34	0.1	33	165	0.4	34	340	0.7	45	180	0.2
(8000,100)	100	1.2	32	0.1	30	150	0.3	36	360	0.6	52	208	0.2
(8000,200)	100	2.4	35	0.1	34	170	0.6	31	310	1.0	45	180	0.4

Table 3: Numerical Results for Lasso problem

$(m, n)$	ISTA		ADMM		ADMM-5			ADMM-10			EGADM		
	iter	cpu	iter	cpu	iter	mvm	cpu	iter	mvm	cpu	iter	mvm	cpu
$\gamma = 0.5$													
(100,1000)	100	0.1	51	2.4	67	335	0.0	54	540	0.0	202	808	0.0
(100,2000)	100	0.3	51	14.6	67	335	0.1	54	540	0.1	202	808	0.1
(100,5000)	100	0.7	51	206.9	67	335	0.4	54	540	0.5	202	808	0.5
(100,8000)	100	1.2	51	793.9	67	335	0.7	54	540	0.9	202	808	0.9
(1000,100)	100	0.1	36	0.0	36	180	0.0	36	360	0.0	95	380	0.0
(1000,200)	100	0.3	46	0.1	42	210	0.0	58	580	0.1	128	512	0.0
(2000,200)	100	0.6	1000	1.3	52	260	0.1	41	410	0.2	126	504	0.2
(5000,100)	100	0.7	33	0.0	31	155	0.1	31	310	0.3	65	260	0.1
(5000,200)	100	1.4	33	0.1	34	170	0.4	35	350	0.7	75	300	0.4
(8000,100)	100	1.2	35	0.1	37	185	0.3	49	490	0.8	71	284	0.3
(8000,200)	100	2.2	34	0.1	37	185	0.6	33	330	1.0	72	288	0.5
$\gamma = 0.1$													
(100,1000)	100	0.1	12	0.5	206	1030	0.1	106	1060	0.1	1000	4000	0.2
(100,2000)	100	0.3	12	3.6	206	1030	0.2	106	1060	0.1	1000	4000	0.4
(100,5000)	100	0.7	12	46.1	206	1030	1.1	106	1060	1.0	1000	4000	2.5
(100,8000)	100	1.2	12	192.3	206	1030	2.1	106	1060	1.9	1000	4000	4.4
(1000,100)	100	0.1	122	0.0	145	725	0.0	109	1090	0.1	592	2368	0.1
(1000,200)	100	0.3	114	0.1	207	1035	0.1	127	1270	0.1	935	3740	0.2
(2000,200)	100	0.6	118	0.2	173	865	0.5	137	1370	0.6	660	2640	0.8
(5000,100)	100	0.8	108	0.1	112	560	0.6	118	1180	1.0	363	1452	0.8
(5000,200)	100	1.4	107	0.2	121	605	1.3	115	1150	2.3	480	1920	2.4
(8000,100)	100	1.2	117	0.1	138	690	1.2	144	1440	2.3	360	1440	1.4
(8000,200)	100	2.4	110	0.3	128	640	2.1	133	1330	4.3	425	1700	3.2