An Extragradient-Based Alternating Direction Method for Convex Minimization

Shiqian MA * Shuzhong ZHANG †

January 26, 2013

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 only one of the two functions has easy proximal mapping, and the other one is smoothly convex but does not have an easy proximal mapping. Therefore, the classical alternating direction methods cannot be applied. For solving this kind of problems, we propose in this paper an alternating direction method based on *extragradients*. Under the assumption that the smooth function has a Lipschitz continuous gradient, we prove that the proposed method returns an ϵ-optimal solution within $O(1/\epsilon)$ iterations. We test the performance of different variants of the proposed method through solving the basis pursuit problem arising from compressed sensing. We then 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.

Keywords: Alternating Direction Method; Extragradient; Iteration Complexity; Basis Pursuit; Fused Logistic Regression

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

*Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, N. T., Hong Kong. Email: sqma@se.cuhk.edu.hk

†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:

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

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 recently 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

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

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 λ 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 [7] and Peaceman-Rachford operator splitting method [29] for finding the zero of the sum of two maximal monotone operators. In particular, it was shown by Gabay [12] 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., [20, 10, 14, 8, 9]). The ADMM was revisited recently because it was found very efficient for solving many sparse and low-rank optimization problems, such as compressed sensing [38], compressive imaging [35, 15], robust PCA [32], sparse inverse covariance selection [41, 30], sparse PCA [22] and semidefinite programming [36] etc. Moreover, the iteration complexity of ADMM (1.2) was recently established by He and Yuan [17] and Monteiro and Svaiter [24]. The recent survey paper by Boyd *et al.* [3] 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 $\tau>0$, $w\in\mathbb{R}^n$ and $z\in\mathbb{R}^p$:

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

and

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

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., [38, 37, 42]) which try to linearize 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:

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

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 ℓ_1 norm of Y. Problem (1.6) was studied by Candès *et al.* [4] and Chandrasekaran *et al.* [5] as a convex relaxation of the robust PCA problem. Note that the two involved functions, the nuclear norm $\|X\|_*$ and the ℓ_1 norm $\|Y\|_1$, have easy proximal mappings (see, e.g., [23] and [16]). The other problem is the so-called sparse inverse covariance selection, which is also known as the graphical lasso problem [40, 1, 11]. This problem, which estimates a sparse inverse covariance matrix from sample data, can be formulated as

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

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

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

Note that the involved function $-\log\det(X)$ has an easy proximal mapping (see, e.g., [30] and [41]).

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

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

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 $\text{Prob}(b \mid a) = 0.5$. Besides, $\text{Prob}(b \mid a) > 0.5$ if $a^\top x + c$ has the same sign as b, and $\text{Prob}(b \mid a) < 0.5$ otherwise. The sparse logistic regression (see [21]) can be formulated as the following convex optimization problem

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

where $\ell(x, c)$ denotes the average logistic loss function, which is defined as

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

and the ℓ_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

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

When ADMM (1.2) is applied to solve (1.10), although the subproblem with respect to y is easily solvable (an ℓ_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:

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

This problem cannot be solved by ADMM (1.2), again because of the difficulty of computing the proximal mapping of $\ell(x, c)$. We will discuss this example in more details in Section 5.

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.

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 ϵ-optimal solution to (1.1) within $O(1/\epsilon)$ iterations. We then compare the performance of some variants of the proposed method using basis pursuit problem from compressed sensing. We also discuss in details the fused logistic regression problem and show that our method can solve this problem effectively.

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.

We propose the following extragradient-based alternating direction method (EGADM) to solve Problem (1.1). Starting with any initial point $y^0 \in \mathcal{Y}$ and $\lambda^0 \in \mathbb{R}^m$, a typical iteration of EGADM can be described as:

$$\left\{\begin{array}{lcl} x^{k+1} & := & \mathrm{argmin}_{x\in\mathcal{X}}\ \mathcal{L}_\gamma(x, y^k; \lambda^k) + \frac{1}{2}\|x - x^k\|_H^2 \\ \bar{y}^{k+1} & := & [y^k - \gamma\nabla_y \mathcal{L}(x^{k+1}, y^k; \lambda^k)]_{\mathcal{Y}} \\ \bar{\lambda}^{k+1} & := & \lambda^k - \gamma(Ax^{k+1} + By^k - b) \\ y^{k+1} & := & [y^k - \gamma\nabla_y \mathcal{L}(x^{k+1}, \bar{y}^{k+1}; \bar{\lambda}^{k+1})]_{\mathcal{Y}} \\ \lambda^{k+1} & := & \lambda^k - \gamma(Ax^{k+1} + B\bar{y}^{k+1} - b), \end{array}\right. \tag{2.1}$$

where $[y]_{\mathcal{Y}}$ denotes the projection of y onto $\mathcal{Y}$, $\mathcal{L}_\gamma(x, y; \lambda)$ is the augmented Lagrangian function for (1.1) as defined in (1.3), H is a pre-specified positive semidefinite matrix, and $\mathcal{L}(x, y; \lambda)$ is the Lagrangian function for (1.1), which is defined as

$$\mathcal{L}(x, y; \lambda) := f(x) + g(y) - \langle\lambda, Ax + By - b\rangle. \tag{2.2}$$

Note that the first subproblem in (2.1) 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.,

$$x^{k+1} := \mathrm{argmin}_{x\in\mathcal{X}}\ f(x) - \langle\lambda^k, Ax + By^k - b\rangle + \frac{\gamma}{2}\|Ax + By^k - b\|^2 + \frac{1}{2}\|x - x^k\|_H^2. \tag{2.3}$$

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 (2.3) is relatively easy to solve. Basically, when A is an identity matrix, and f is a function arising from sparse optimization such as ℓ_1 norm, ℓ_2 norm, nuclear norm etc, the subproblem (2.3) is usually easy to solve.

Because we do not impose any structure on the smooth convex function g, the following subproblem is not assumed to be easily solvable to optimality:

$$\min_{y\in\mathcal{Y}}\ \mathcal{L}_\gamma(x,y;\lambda) \tag{2.4}$$

for given x and λ. Thus, in the extragradient-based ADM (2.1), we do not solve (2.4). Instead, we take gradient projection steps for Lagrangian function $\mathcal{L}(x,y;\lambda)$ for fixed x and λ to update y. Note that since the gradient of $\mathcal{L}(x,y;\lambda)$ with respect to λ is given by

$$\nabla_\lambda\mathcal{L}(x,y;\lambda)=-(Ax+By-b), \tag{2.5}$$

the two updating steps for λ in (2.1) can be interpreted as

$$\bar\lambda^{k+1}:=\lambda^k+\gamma\nabla_\lambda\mathcal{L}(x^{k+1},y^k;\lambda^k) \tag{2.6}$$

and

$$\lambda^{k+1}:=\lambda^k+\gamma\nabla_\lambda\mathcal{L}(x^{k+1},\bar y^{k+1};\bar\lambda^{k+1}). \tag{2.7}$$

Hence, 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 (2.1) equivalently as

$$\left\{\begin{array}{lcl} x^{k+1} &:=& \mathrm{argmin}_{x\in\mathcal{X}}\mathcal{L}_\gamma(x,y^k;\lambda^k)+\frac{1}{2}\|x-x^k\|_H^2\\ \bar z^{k+1} &:=& [z^k-\gamma F(x^{k+1},z^k)]_{\mathcal{Z}}\\ z^{k+1} &:=& [z^k-\gamma F(x^{k+1},\bar z^{k+1})]_{\mathcal{Z}}.\end{array}\right. \tag{2.8}$$

Now it is easy to see that the two steps for updating z in (2.8) are gradient projection steps for the Lagrangian function $\mathcal{L}(x^{k+1},y;\lambda)$ for (y^k,λ^k) and $(\bar y^{k+1},\bar\lambda^{k+1})$ respectively. The steps for y are gradient descent steps and the steps for λ are gradient ascent steps, because the original problem (1.1) can be rewritten as a minimax problem:

$$\min_{x\in\mathcal{X},y\in\mathcal{Y}}\max_\lambda\ f(x)+g(y)-\langle\lambda,Ax+By-b\rangle, \tag{2.9}$$

which minimizes the Lagrangian function with respect to x and y but maximizes the Lagrangian function with respect to λ. Since there are two gradient steps for updating z, the second gradient step can be seen as an extragradient step. 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 [18, 19]. Korpelevich proved the convergence of the extragradient method [18, 19]. For recent results on convergence of extragradient type methods, we refer to [28] and the references therein. The iteration complexity result for extragradient method was analyzed by Nemirovski in [27]. Recently, Monteiro and Svaiter [25, 26, 24] studied the iteration complexity results of the hybrid proximal extragradient method proposed by Solodov and Svaiter in [31] and its variants. More recently, Bonettini and Ruggiero studied a generalized extragradient method for total variation based image restoration problem [2].

3 Iteration Complexity

In this section, we analyze the iteration complexity of EGADM, i.e., (2.1). We show that under the assumption that the smooth function g has a Lipschitz continuous gradient, EGADM (2.1) finds an ϵ-optimal solution (defined in Definition 3.1) to Problem (1.1) within $O(1/\epsilon)$ iterations. The Lagrangian dual problem of (1.1) is

$$\max_{\lambda} d(\lambda), \tag{3.1}$$

where

$$d(\lambda) = \min_{x\in\mathcal{X}, y\in\mathcal{Y}} \mathcal{L}(x, y; \lambda).$$

The ϵ-optimal solution to Problem (1.1) is thus defined as follows.

Definition 3.1 *We call $(\hat{x}, \hat{y}) \in \mathcal{X} \times \mathcal{Y}$ and $\hat{\lambda} \in \mathbb{R}^m$ a pair of ϵ-optimal solution to Problem (1.1), if the following holds,*

$$\max_{x\in\mathcal{X}^\star, y\in\mathcal{Y}^\star, \lambda\in\Lambda^\star} \left(\mathcal{L}(\hat{x}, \hat{y}; \lambda) - \mathcal{L}(x, y; \hat{\lambda})\right) \leq \epsilon, \tag{3.2}$$

where $\mathcal{X}^\star \times \mathcal{Y}^\star$ is the optimal set of the primal problem (1.1) and $\Lambda^\star$ is the optimal set of the dual problem (3.1).

The ϵ-optimal solution defined in Definition 3.1 measures the closeness of the optimal solution to the optimal set in terms of the duality gap. This is validated by the following saddle point theorem.

Theorem 3.1 *(Saddle-Point Theorem) The pair $(x^\star, y^\star; \lambda^\star)$ with $(x^\star, y^\star) \in \mathcal{X} \times \mathcal{Y}$ and $\lambda^\star \in \mathbb{R}^m$ is a primal-dual optimal solution pair to (1.1) if and only if $(x^\star, y^\star)$ and $\lambda^\star$ satisfy*

$$\mathcal{L}(x^\star, y^\star; \lambda) \leq \mathcal{L}(x^\star, y^\star; \lambda^\star) \leq \mathcal{L}(x, y; \lambda^\star), \quad \textit{for all } x \in \mathcal{X}, y \in \mathcal{Y}, \lambda \in \mathbb{R}^m,$$

i.e., $(x^\star, y^\star; \lambda^\star)$ *is a saddle point of the Lagrangian function* $\mathcal{L}(x, y; \lambda)$.

It is well known that the weak duality holds for the primal problem (1.1) and the dual problem (3.1), i.e., $d(\lambda) \leq f(x) + g(y)$ for any feasible solutions (x, y) and λ. In our particular case, the Lagrangian dual variable λ is associated with a set of *linear equality* constraints. The strong duality holds if the primal problem has an optimal solution. Furthermore, we assume that the optimal set $(\mathcal{X}^\star, \mathcal{Y}^\star)$ to the primal problem (1.1) and the optimal set $\Lambda^\star$ to the dual problem (3.1) are both bounded. This assumption indeed holds for a wide variety of problem classes (e.g. when the primal objective function is coercive and continuously differentiable). Now we are ready to analyze the iteration complexity of EGADM (2.8), or equivalently, (2.1), for an ϵ-optimal solution in the sense of Definition 3.1. We will prove the following lemma first.

Lemma 3.2 *The sequence* $\{x^{k+1}, z^k, \bar{z}^k\}$ *generated by* (2.8) *satisfies the following inequality:*

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

Proof. Note that the optimality conditions of the two subproblems for z in (2.8) are given by

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

and

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

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

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

which implies

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

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

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

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. □

We next prove the following lemma.

Lemma 3.3 *Assume that $\nabla g(y)$ is Lipschitz continuous with Lipschitz constant L_g, i.e.,*

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

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

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

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,

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

which combining with (3.3) yields (3.10). □

We further prove the following lemma.

Lemma 3.4 *Under the same assumptions as in Lemma 3.3, the following holds:*

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

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,

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

where the last inequality is due to (3.10). □

We now give the $O(1/\epsilon)$ iteration complexity of (2.1) (or equivalently, (2.8)) for an ϵ-optimal solution to Problem (1.1).

Theorem 3.5 *Consider Algorithm EGADM (2.1), and its sequence of iterates. For any integer $N > 0$, define*

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

Suppose that the optimal solution sets of the primal problem (1.1) and the dual problem (3.1) are both bounded. Moreover, assume that $\nabla g(y)$ is Lipschitz continuous with Lipschitz constant L_g, and we choose $\gamma \leq 1/(2\hat{L})$, where $\hat{L} := \left(\max\{2L_g^2 + \lambda_{\max}(B^\top B), 2\lambda_{\max}(B^\top B)\}\right)^{\frac{1}{2}}$. 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 $\tau > \gamma\lambda_{\max}(A^\top A)$. We have the following inequalities:

$$\max_{x \in \mathcal{X}^\star, y \in \mathcal{Y}^\star, \lambda \in \Lambda^\star} \left(\mathcal{L}(\tilde{x}^N, \tilde{y}^N; \lambda) - \mathcal{L}(x, y; \tilde{\lambda}^N) \right) \leq \frac{1}{2\gamma N} \max_{z \in \mathcal{Z}^\star} \|z - z^0\|^2 + \frac{1}{2N} \max_{x \in \mathcal{X}^\star} \|x - x^0\|^2,$$

where $\mathcal{Z}^\star := \mathcal{Y}^\star \times \Lambda^\star$. This implies that when $N = O(1/\epsilon)$, $\{\tilde{x}^N, \tilde{y}^N, \tilde{\lambda}^N\}$ is an ϵ-optimal solution to Problem (1.1); i.e., the iteration complexity of (2.1) (or equivalently, (2.8)) for an ϵ-optimal solution to Problem (1.1) is $O(1/\epsilon)$.

Proof. The optimality conditions of the subproblem for x in (2.1) are given by

$$\langle \partial f(x^{k+1}) - A^\top \lambda^k + \gamma A^\top (Ax^{k+1} + By^k - b) + H(x^{k+1} - x^k), x - x^{k+1} \rangle \geq 0, \quad \forall x \in \mathcal{X}. \quad (3.14)$$

By using the updating formula for $\bar{\lambda}^{k+1}$ in (2.1), 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,

$$\langle \partial f(x^{k+1}) - A^\top \bar{\lambda}^{k+1} + H(x^{k+1} - x^k), x - x^{k+1} \rangle \geq 0, \quad \forall x \in \mathcal{X}. \tag{3.15}$$

Combining (3.11) and (3.15), we have,

$$\begin{aligned} & \left\langle \begin{pmatrix} x - x^{k+1} \\ y - \bar{y}^{k+1} \\ \lambda - \bar{\lambda}^{k+1} \end{pmatrix}, \begin{pmatrix} \partial f(x^{k+1}) - A^\top \bar{\lambda}^{k+1} \\ \nabla g(\bar{y}^{k+1}) - B^\top \bar{\lambda}^{k+1} \\ Ax^{k+1} + B\bar{y}^{k+1} - b \end{pmatrix} \right\rangle \\ \geq \; & \tfrac{1}{2\gamma} \left(\|z - z^{k+1}\|^2 - \|z - z^k\|^2 \right) + \langle H(x^k - x^{k+1}), x - x^{k+1} \rangle. \end{aligned} \tag{3.16}$$

Since

$$\langle H(x^k - x^{k+1}), x - x^{k+1} \rangle = -\frac{1}{2}\|x^k - x\|_H^2 + \frac{1}{2}\|x^k - x^{k+1}\|_H^2 + \frac{1}{2}\|x - x^{k+1}\|_H^2 \geq \frac{1}{2}\|x - x^{k+1}\|_H^2 - \frac{1}{2}\|x - x^k\|_H^2,$$

summing (3.16) over $k = 0, 1, \ldots, N$ yields,

$$\begin{aligned} & -\tfrac{1}{2\gamma N}\|z - z^0\|^2 - \tfrac{1}{2N}\|x - x^0\|_H^2 \\ \leq \; & \tfrac{1}{N}\sum_{k=1}^N \left\langle \begin{pmatrix} \nabla_x \mathcal{L}(x^{k+1}, \bar{y}^{k+1}, \bar{\lambda}^{k+1}) \\ \nabla_y \mathcal{L}(x^{k+1}, \bar{y}^{k+1}, \bar{\lambda}^{k+1}) \end{pmatrix}, \begin{pmatrix} x - x^{k+1} \\ y - \bar{y}^{k+1} \end{pmatrix} \right\rangle \\ & + \tfrac{1}{N}\sum_{k=1}^N \langle -\nabla_\lambda \mathcal{L}(x^{k+1}, \bar{y}^{k+1}, \bar{\lambda}^{k+1}), \lambda - \bar{\lambda}^{k+1} \rangle \\ \leq \; & \tfrac{1}{N}\sum_{k=1}^N \left(\mathcal{L}(x, y, \bar{\lambda}^{k+1}) - \mathcal{L}(x^{k+1}, \bar{y}^{k+1}, \bar{\lambda}^{k+1}) \right) + \tfrac{1}{N}\sum_{k=1}^N \left(\mathcal{L}(x^{k+1}, \bar{y}^{k+1}, \bar{\lambda}^{k+1}) - \mathcal{L}(x^{k+1}, \bar{y}^{k+1}, \lambda) \right) \\ = \; & \tfrac{1}{N}\sum_{k=1}^N \left(\mathcal{L}(x, y, \bar{\lambda}^{k+1}) - \mathcal{L}(x^{k+1}, \bar{y}^{k+1}, \lambda) \right) \\ \leq \; & \mathcal{L}(x, y, \tfrac{1}{N}\sum_{k=1}^N \bar{\lambda}^{k+1}) - \mathcal{L}(\tfrac{1}{N}\sum_{k=1}^N x^{k+1}, \tfrac{1}{N}\sum_{k=1}^N \bar{y}^{k+1}; \lambda) \\ = \; & \mathcal{L}(x, y; \tilde{\lambda}^N) - \mathcal{L}(\tilde{x}^N, \tilde{y}^N; \lambda). \end{aligned}$$

Thus we have,

$$\begin{aligned} & \max_{x \in \mathcal{X}^\star, y \in \mathcal{Y}^\star, \lambda \in \Lambda^\star} \left(\mathcal{L}(\tilde{x}^N, \tilde{y}^N; \lambda) - \mathcal{L}(x, y; \tilde{\lambda}^N) \right) \\ = \; & \left(\max_{\lambda \in \Lambda^\star} \mathcal{L}(\tilde{x}^N, \tilde{y}^N; \lambda) - \min_{x \in \mathcal{X}^\star, y \in \mathcal{Y}^\star} \max_{\lambda \in \Lambda^\star} \mathcal{L}(x, y; \lambda) \right) \\ & + \left(\max_{\lambda \in \Lambda^\star} \min_{x \in \mathcal{X}^\star, y \in \mathcal{Y}^\star} \mathcal{L}(x, y; \lambda) - \min_{x \in \mathcal{X}^\star, y \in \mathcal{Y}^\star} \mathcal{L}(x, y; \tilde{\lambda}^N) \right) \\ \leq \; & \tfrac{1}{2\gamma N} \max_{z \in \mathcal{Z}^\star} \|z - z^0\|^2 + \tfrac{1}{2N} \max_{x \in \mathcal{X}^\star} \|x - x^0\|^2. \end{aligned}$$

Because $\mathcal{X}^\star$, $\mathcal{Y}^\star$ and $\Lambda^\star$ are all bounded, $N = O(1/\epsilon)$ will guarantee that

$$\max_{x \in \mathcal{X}^\star, y \in \mathcal{Y}^\star, \lambda \in \Lambda^\star} \left(\mathcal{L}(\tilde{x}^N, \tilde{y}^N; \lambda) - \mathcal{L}(x, y; \tilde{\lambda}^N) \right) \leq \epsilon,$$

and this completes the proof. □

4 Variants of EGADM

In this section, we propose some variants of EGADM (2.1). Note that in (2.1), we used the augmented Lagrangian function in the subproblem with respect to x, while we used the Lagrangian

function in computing the gradient steps for y and $\bar{y}$. So one variant of (2.1) we will propose in this section is to use the augmented Lagrangian function in computing the gradient steps for y and $\bar{y}$, and it can be described as follows:

$$\begin{cases} x^{k+1} & := & \operatorname{argmin}_{x\in\mathcal{X}} \mathcal{L}_\gamma(x, y^k; \lambda^k) + \frac{1}{2}\|x - x^k\|_H^2 \\ \bar{y}^{k+1} & := & [y^k - \gamma\nabla_y \mathcal{L}_\gamma(x^{k+1}, y^k; \lambda^k)]_{\mathcal{Y}} \\ \bar{\lambda}^{k+1} & := & \lambda^k - \gamma(Ax^{k+1} + By^k - b) \\ y^{k+1} & := & [y^k - \gamma\nabla_y \mathcal{L}_\gamma(x^{k+1}, \bar{y}^{k+1}; \bar{\lambda}^{k+1})]_{\mathcal{Y}} \\ \lambda^{k+1} & := & \lambda^k - \gamma(Ax^{k+1} + B\bar{y}^{k+1} - b). \end{cases} \tag{4.1}$$

Note that the gradient steps for λ and $\bar{\lambda}$ are unchanged, because the gradients of the Lagrangian function and the augmented Lagrangian function with respect to λ are the same. By dropping the extra gradient steps in (2.1) and (4.1), we get the following two other variants of EGADM:

$$\begin{cases} x^{k+1} & := & \operatorname{argmin}_{x\in\mathcal{X}} \mathcal{L}_\gamma(x, y^k; \lambda^k) + \frac{1}{2}\|x - x^k\|_H^2 \\ y^{k+1} & := & [y^k - \gamma\nabla_y \mathcal{L}(x^{k+1}, y^k; \lambda^k)]_{\mathcal{Y}} \\ \lambda^{k+1} & := & \lambda^k - \gamma(Ax^{k+1} + By^{k+1} - b), \end{cases} \tag{4.2}$$

and

$$\begin{cases} x^{k+1} & := & \operatorname{argmin}_{x\in\mathcal{X}} \mathcal{L}_\gamma(x, y^k; \lambda^k) + \frac{1}{2}\|x - x^k\|_H^2 \\ y^{k+1} & := & [y^k - \gamma\nabla_y \mathcal{L}_\gamma(x^{k+1}, y^k; \lambda^k)]_{\mathcal{Y}} \\ \lambda^{k+1} & := & \lambda^k - \gamma(Ax^{k+1} + By^{k+1} - b). \end{cases} \tag{4.3}$$

We let "GL" denote the ADM based on gradient for Lagrangian function (4.2), "GAL" denote the ADM based on gradient for augmented Lagrangian function (4.3), "EGL" denote the ADM based on extra gradient for Lagrangian function (2.1), and "EGAL" denote the ADM based on extra gradient for augmented Lagrangian function (4.1). For the performance of these four algorithms, we have the following observations based on our numerical results in Section 6. The performance of "GL" is the worst among these four algorithms. It usually does not converge or converges very slowly. The performance of "EGAL" is the best among these four algorithms. There is no significant difference between the performance of "GAL" and "EGL", and they usually perform worse than "EGAL" but better than "GL". Although at this point we are only able to prove the convergence and iteration complexity for "EGL", i.e. (2.1), our numerical results suggest that "EGAL", i.e. (4.1), should have at least the same convergence properties, which presently remains an unsolved problem.

5 Fused Logistic Regression

In this section, we show how to use Algorithm (2.1) 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 [33], was introduced to find sparse regression coefficients so that the resulting model is more interpretable. The original Lasso model solves the following problem:

$$\min \ \frac{1}{2}\|Ax-b\|^2, \ \text{s.t.} \ \|x\|_1 \leq s, \tag{5.1}$$

where $A = [a_1, \ldots, a_m]^\top \in \mathbb{R}^{m\times n}$ gives the predictor variables, $b = [b_1, \ldots, b_m]^\top \in \mathbb{R}^m$ 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 [34] to model the situation that 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

$$\min \ \frac{1}{2}\|Ax-b\|^2 + \alpha\|x\|_1 + \beta\sum_{j=2}^{n}|x_j - x_{j-1}|. \tag{5.2}$$

Because (5.2) can be transformed equivalently to a quadratic programming problem, Tibshirani *et al.* proposed to solve (5.2) using a two-phase active set algorithm SQOPT of Gill *et al.* [13]. However, transforming (5.2) to a quadratic programming problem will increase the size of the problem significantly, thus SQOPT can only solve (5.2) with small or medium sizes. Ye and Xie [39] proposed to solve (5.2) using split Bregman algorithm, which can be shown to be equivalent to an alternating direction method of multipliers. Note that (5.2) can be rewritten equivalently as

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax-b\|^2 + \alpha\|w\|_1 + \beta\|y\|_1 \\ \text{s.t.} & w = x \\ & y = Lx, \end{array} \tag{5.3}$$

where L is an $(n-1)\times n$ dimensional matrix with all ones in the diagonal and negative ones in the super-diagonal and zeros elsewhere. The ADMM (split Bregman) for solving (5.3) can be described as

$$\left\{\begin{array}{lcl} x^{k+1} & := & \text{argmin}_x \ \mathcal{L}_\gamma(x, w^k, y^k; \lambda_1^k, \lambda_2^k) \\ (w^{k+1}, y^{k+1}) & := & \text{argmin}_{w,y} \ \mathcal{L}_\gamma(x^{k+1}, w, y; \lambda_1^k, \lambda_2^k) \\ \lambda_1^{k+1} & := & \lambda_1^k - \gamma(w^{k+1} - x^{k+1}) \\ \lambda_2^{k+1} & := & \lambda_2^k - \gamma(y^{k+1} - Lx^{k+1}), \end{array}\right. \tag{5.4}$$

where

$$\mathcal{L}_\gamma(x, w, y; \lambda_1, \lambda_2) := \frac{1}{2}\|Ax-b\|^2 + \alpha\|w\|_1 + \beta\|y\|_1 - \langle\lambda_1, w-x\rangle - \langle\lambda_2, y-Lx\rangle + \frac{\gamma}{2}\|w-x\|^2 + \frac{\gamma}{2}\|y-Lx\|^2$$

is the augmented Lagrangian function for (5.3), λ_1 and λ_2 are Lagrange multipliers associated with the two constraints. Note that the subproblem for x in (5.4) requires to minimize a positive-define

quadratic function, which can be time consuming, especially when the size of A is large. The subproblem for w and y corresponds to the ℓ_1 shrinkage operations that can be given in analytical form.

As in [21], the sparse logistic regression problem (1.9) can also be formulated as

$$\min_{x,c} \ \ell(x,c), \ \text{s.t.} \ \|x\|_1 \le s. \tag{5.5}$$

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:

$$\min_{x\in\mathbb{R}^n, c\in\mathbb{R}} \ \ell(x,c) + \alpha\|x\|_1 + \beta\sum_{j=2}^{n}|x_j - x_{j-1}|. \tag{5.6}$$

Problem (5.6) can be rewritten equivalently as

$$\begin{array}{ll} \min_{x\in\mathbb{R}^n, w\in\mathbb{R}^{n-1}, y\in\mathbb{R}^n, c\in\mathbb{R}} & \alpha\|x\|_1 + \beta\|w\|_1 + \ell(y,c) \\ \text{s.t.} & x = y \\ & w = Ly. \end{array} \tag{5.7}$$

If we apply the ADMM as in (5.4) to solve (5.7), we will end up with the following iterates:

$$\left\{\begin{array}{lcl} (x^{k+1}, w^{k+1}) & := & \text{argmin}_{x,w} \ \mathcal{L}_\gamma(x,w,y^k,c^k;\lambda_1^k,\lambda_2^k) \\ (y^{k+1}, c^{k+1}) & := & \text{argmin}_{y,c} \ \mathcal{L}_\gamma(x^{k+1},w^{k+1},y,c;\lambda_1^k,\lambda_2^k) \\ \lambda_1^{k+1} & := & \lambda_1^k - \gamma(x^{k+1} - y^{k+1}) \\ \lambda_2^{k+1} & := & \lambda_2^k - \gamma(w^{k+1} - Ly^{k+1}), \end{array}\right. \tag{5.8}$$

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

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

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 extragradient-based ADM to solve (5.7). Based on the discussions in Section 4, we know that "EGAL", i.e., ADM with extragradients for augmented Lagrangian function, usually performs the best. We thus only show the details of "EGAL" for solving (5.7) in (5.9). One can easily get the corresponding "EGL" for solving (5.7) just by replacing the augmented

Lagrangian function by the Lagrangian function when computing the gradients for y steps:

$$\left\{\begin{array}{lcl} (x^{k+1}, w^{k+1}) & := & \mathrm{argmin}_{x,w}\ \mathcal{L}_\gamma(x, w, y^k, c^k; \lambda_1^k, \lambda_2^k) \\ \bar{y}^{k+1} & := & y^k - \gamma \nabla_y \mathcal{L}_\gamma(x^{k+1}, w^{k+1}, y^k, c^k; \lambda_1^k, \lambda_2^k) \\ \bar{c}^{k+1} & := & c^k - \gamma \nabla_c \mathcal{L}_\gamma(x^{k+1}, w^{k+1}, y^k, c^k; \lambda_1^k, \lambda_2^k) \\ \bar{\lambda}_1^{k+1} & := & \lambda_1^k - \gamma(x^{k+1} - y^k) \\ \bar{\lambda}_2^{k+1} & := & \lambda_2^k - \gamma(w^{k+1} - L y^{k+1}) \\ y^{k+1} & := & y^k - \gamma \nabla_y \mathcal{L}_\gamma(x^{k+1}, w^{k+1}, \bar{y}^{k+1}, \bar{c}^{k+1}; \bar{\lambda}_1^{k+1}, \bar{\lambda}_2^{k+1}) \\ c^{k+1} & := & c^k - \gamma \nabla_c \mathcal{L}_\gamma(x^{k+1}, w^{k+1}, \bar{y}^{k+1}, \bar{c}^{k+1}; \bar{\lambda}_1^{k+1}, \bar{\lambda}_2^{k+1}) \\ \lambda_1^{k+1} & := & \lambda_1^k - \gamma(x^{k+1} - \bar{y}^{k+1}) \\ \lambda_2^{k+1} & := & \lambda_2^k - \gamma(w^{k+1} - L\bar{y}^{k+1}). \end{array}\right. \tag{5.9}$$

Now we show how to compute the updates in (5.9) explicitly. First, the subproblem for (x, w) can be reduced to the following two subproblems:

$$x^{k+1} := \mathrm{argmin}_x\ \frac{\alpha}{\gamma}\|x\|_1 + \frac{1}{2}\left\|x - \left(y^k + \frac{\lambda_1^k}{\gamma}\right)\right\|^2 \tag{5.10}$$

and

$$w^{k+1} := \mathrm{argmin}_w\ \frac{\beta}{\gamma}\|w\|_1 + \frac{1}{2}\left\|w - \left(Ly^k + \frac{\lambda_2^k}{\gamma}\right)\right\|^2. \tag{5.11}$$

These two subproblems have closed-form solutions given by

$$x^{k+1} := \mathrm{Shrink}(y^k + \lambda_1^k/\gamma, \alpha/\gamma) \tag{5.12}$$

and

$$w^{k+1} := \mathrm{Shrink}(Ly^k + \lambda_2^k/\gamma, \beta/\gamma), \tag{5.13}$$

where the ℓ_1 shrinkage operator $\mathrm{Shrink}(z, \tau)$ is defined as

$$\mathrm{Shrink}(z, \tau) := \mathrm{sign}(z) \circ \max\{|z| - \tau, 0\}. \tag{5.14}$$

To compute the updates for y and c, we need to compute the gradients of $\ell(y, c)$ with respect to y and c. It is easy to check that they can be obtained by:

$$\nabla_y \ell(y, c) = -\frac{1}{m}\hat{A}^\top(1 - d),\ \nabla_c \ell(y, c) = -\frac{1}{m} b^\top(1 - d),\ d = 1./(1 + \exp(-\hat{A}y - b \circ c)), \tag{5.15}$$

where $\hat{A} = [b_1 a_1, b_2 a_2, \ldots, b_m a_m]^\top$.

Based on these discussions, we can summarize the extragradient-based ADM for solving (5.7) as Algorithm 1.

Algorithm 1 Extragradient-based ADM for the Fused Logistic Regression

Initialization: $\hat{A} = [b_1 a_1, b_2 a_2, \ldots, b_m a_m]^\top$

for $k = 0, 1, \ldots$ **do**

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

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

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

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

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

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

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

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

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

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

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

end for

6 Numerical Experiments

In this section, we test the performance of our extragradient-based ADM for solving basis pursuit problem arising from compressed sensing and the fused logistic regression problem (5.6). 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.

6.1 Numerical Results for Basis Pursuit

The cardinality minimization problem arising from compressed sensing seeks the sparsest solution of a linear system. It can be formulated as

$$\min_{x \in \mathbb{R}^n} \ \|x\|_0, \ \text{s.t.}, \ Ax = b, \tag{6.1}$$

where $A \in \mathbb{R}^{m \times n}$ (without loss of generality, we assume A has full row rank), $b \in \mathbb{R}^m$ and $\|x\|_0$ counts the number of nonzeros of x. The theory of compressed sensing shows that under certain conditions, the cardinality minimization problem (6.1) is equivalent to the following ℓ_1 minimization problem with high probability:

$$\min_{x \in \mathbb{R}^n} \ \|x\|_1, \ \text{s.t.}, \ Ax = b. \tag{6.2}$$

Problem (6.2) is also known as the basis pursuit problem [6].

In this section, we show the performance of the four algorithms discussed in Section 4, i.e., "GL",

"GAL", "EGL" and "EGAL", by solving the basis pursuit problem (6.2). There are many existing efficient algorithms for solving (6.2). We are not trying to compare our algorithms with the existing methods for solving (6.2). We use (6.2) only to illustrate the performance of the four algorithms "GL", "GAL", "EGL" and "EGAL". We show in the following how we use "EGL", i.e., (2.1) to solve (6.2). The other three algorithms are similarly implemented. First, we rewrite (6.2) as

$$\begin{array}{ll} \min_{x\in\mathbb{R}^n, y\in\mathbb{R}^n} & \|x\|_1 \\ \text{s.t.} & x - y = 0, \\ & y \in \mathcal{Y}, \end{array} \tag{6.3}$$

where $\mathcal{Y} := \{y \mid Ay = b\}$. The Lagrangian function and the augmented Lagrangian function for (6.3) are

$$\mathcal{L}(x, y; \lambda) := \|x\|_1 - \langle \lambda, x - y\rangle, \tag{6.4}$$

and

$$\mathcal{L}_\gamma(x, y; \lambda) := \|x\|_1 - \langle \lambda, x - y\rangle + \frac{\gamma}{2}\|x - y\|^2. \tag{6.5}$$

Then the extragradient-based ADM (2.1) for solving (6.3) can be described as

$$\left\{ \begin{array}{lcl} x^{k+1} & := & \operatorname{argmin}_x \ \mathcal{L}_\gamma(x, y^k; \lambda^k) \\ \bar{y}^{k+1} & := & [y^k - \gamma\lambda^k]_{\mathcal{Y}} \\ \bar{\lambda}^{k+1} & := & \lambda^k - \gamma(x^{k+1} - y^k) \\ y^{k+1} & := & [y^k - \gamma\bar{\lambda}^{k+1}]_{\mathcal{Y}} \\ \lambda^{k+1} & := & \lambda^k - \gamma(x^{k+1} - \bar{y}^{k+1}). \end{array} \right. \tag{6.6}$$

Note that we have chosen $H = 0$. It is easy to see that solving the subproblem with respect to x corresponds to the ℓ_1 shrinkage operation. Computing the projection onto $\mathcal{Y}$ can be done by solving a linear system, i.e.,

$$[w]_{\mathcal{Y}} := w + A^\top(AA^\top)^{-1}(b - Aw).$$

The problem instances of (6.2) in our tests were generated in the following manner. For each set of parameters (m, n, s), where s denotes the cardinality of the sparse solution $\hat{x}$ of (6.2), we created ten problem instances. The entries of A were drawn randomly according to Gaussian distribution $\mathcal{N}(0, 1)$. We then normalized A by setting $A := A/\|A\|$, where $\|A\|$ denotes the largest singular value of A. To generate $\hat{x}$, we first chose the locations of the s nonzero components of $\hat{x}$ uniformly random, and then drew the nonzero values of the s components uniformly random in $(0, 1)$. Finally we set $b = A\hat{x}$. We terminated the four algorithms "GL", "GAL", "EGL" and "EGAL" whenever $\|x^k - \bar{y}^k\|_2 < 10^{-4}$. We also terminated the algorithms whenever the iteration number exceeds 20000. The comparison results of the four algorithms are reported in Table 1. In Table 1, "Iter" denotes the number of iterations, and "T" denotes the CPU time in seconds.

From Table 1 we have the following observations. "GL" was the worst one among the four algorithms. It always achieved the maximum iteration number and the error between x^k and $\hat{x}$ was usually high. The other three algorithms can all converge and meet the stopping criterion within a reasonable number of iterations. Especially, the performance of "GAL" and "EGL" is similar as they required similar number of iterations to meet the stopping criterion and generated solutions with similar errors to $\hat{x}$. "EGAL" was the best one among the four algorithms. It needed the least iteration number to meet the stopping criterion and produced solutions with similar errors compared to "GAL" and "EGL". In terms of CPU times, "GAL" and "EGAL" were comparable, and "EGL" usually needed more CPU times than "GAL" and "EGAL". Thus, from these tests on basis pursuit problems, we see that "EGAL" usually performed better than "EGL". Although our current convergence proof and iteration complexity analysis only apply for "EGL", we believe that "EGAL" should have at least the same convergence and complexity results. This will remain as a future research topic.

6.2 Numerical Results for Fused Logistic Regression

In this section, we report the results of our extragradient-based ADM (Algorithm 1) for solving the fused logistic regression problem (5.6).

First, we used a very simple example to show that when the features have natural ordering, the fused logistic regression model (5.6) is much preferable than the sparse logistic regression model (5.5). This simple example was created in the following manner. We created the regression coefficient $\hat{x} \in \mathbb{R}^n$ for $n = 1000$ as

$$\hat{x}_j = \begin{cases} r_1, & j = 1, 2, \ldots, 100 \\ r_2, & j = 201, 202, \ldots, 300 \\ r_3, & j = 401, 402, \ldots, 500 \\ r_4, & j = 601, 602, \ldots, 700 \\ 0, & \text{else}, \end{cases} \tag{6.7}$$

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 (5.6) and compared the result with the sparse logistic regression problem (5.5). The codes for solving (5.5), which is called Lassplore and proposed by Liu *et al.* in [21], were downloaded from http://www.public.asu.edu/∼jye02/Software/lassplore/. Default settings of Lassplore were used. We chose $\alpha = 5 \times 10^{-4}$ and $\beta = 5 \times 10^{-2}$ in (5.6). The regression result by EGADM (Algorithm 1) is plotted in Figure 2 (a). We tested different choices of $s = 1, 5, 10$ in (5.5) and the results are

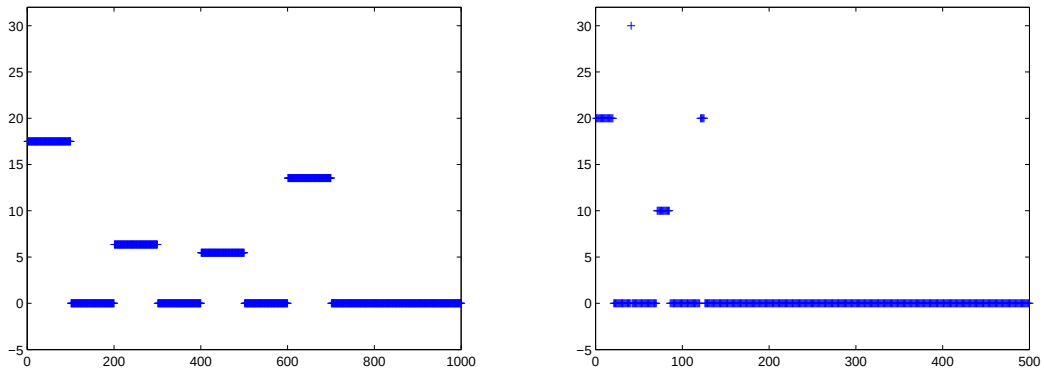

Figure 1: Left: The regression coefficient given in (6.7); Right: The regression coefficient given in (6.8).

plotted in Figure 2 (b), (c) and (d), respectively. From Figure 2 we see that, the fused logistic regression model (5.6) can preserve the natural ordering very well. The sparse logistic regression model (5.5) 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 (5.6), 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

$$\hat{x}_j = \begin{cases} 20, & j = 1, 2, \ldots, 20, \\ 30, & j = 41, \\ 10, & j = 71, \ldots, 85, \\ 20, & j = 121, \ldots, 125, \\ 0, & \text{else.} \end{cases} \tag{6.8}$$

Note that a similar test example was used in [39] 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 (5.6) 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 2. From Table 2 we see that our EGADM can solve the fused logistic regression problem (5.6) efficiently. It solved instances with size up to $m = 2000$, $n = 20000$ in just a few seconds.

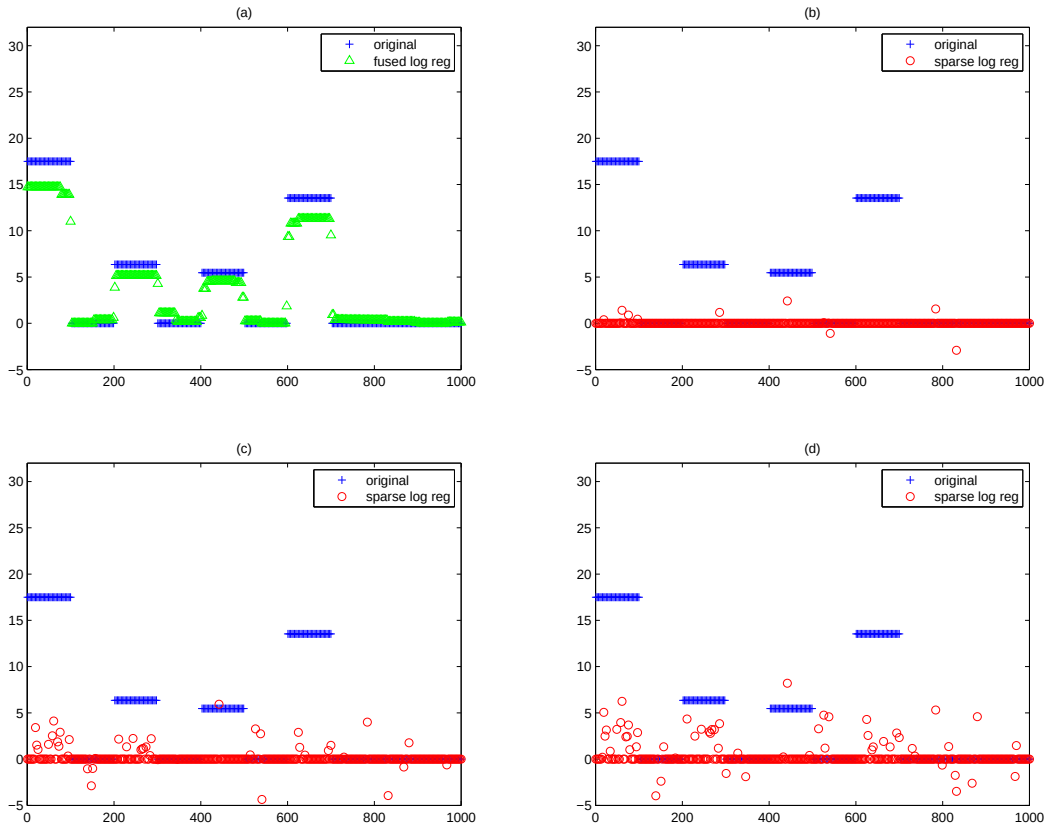

Figure 2: (a): The regression result by the fused logistic regression model (5.6); (b), (c), (d): The regression result by the sparse logistic regression model (5.5) with $s = 1, 5, 10$, respectively.

7 Conclusion

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 ϵ-optimal solution within $O(1/\epsilon)$ iterations. We used the basis pursuit problem to illustrate the performance of the proposed method and its variants. 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.

Acknowledgements

Shiqian Ma thanks Lingzhou Xue and Hui Zou for fruitful discussions on logistic regression and fused lasso.

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. 3

[2] S. Bonettini and V. Ruggiero. An alternating extragradient method for total variation based image restoration from Poisson data. *Inverse Problems*, 27:095001, 2011. 7

[3] 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. 2

[4] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of ACM*, 58(3):1–37, 2011. 3

[5] V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011. 3

[6] S. Chen, D. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998. 16

[7] 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. 2

[8] J. Eckstein. *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis, Massachusetts Institute of Technology, 1989. 2

[9] 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. 2

[10] M. Fortin and R. Glowinski. *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems*. North-Holland Pub. Co., 1983. 2

[11] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008. 3

[12] 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-Hollan, Amsterdam, 1983. 2

[13] 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. 13

[14] R. Glowinski and P. Le Tallec. *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM, Philadelphia, Pennsylvania, 1989. 2

[15] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.*, 2:323–343, 2009. 2

[16] E. T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for ℓ_1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008. 3

[17] 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. 2

[18] 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). 7

[19] G. Korpelevich. Extrapolation gradient methods and relation to modified lagrangeans. *Ekonomika i Matematicheskie Metody*, 19:694–703, 1983. (in Russian; English translation in Matekon). 7

[20] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16:964–979, 1979. 2

[21] J. Liu, J. Chen, and J. Ye. Large-scale sparse logistic regression. In *SIGKDD*, 2009. 4, 14, 18

[22] S. Ma. Alternating direction method of multipliers for sparse principal component analysis. *preprint*, 2011. 2

[23] 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. 3

[24] R. D. C. Monteiro and B. F. Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *Preprint*, 2010. 2, 7

[25] 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. 7

[26] 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. 7

[27] 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. 7

[28] M. A. Noor. New extragradient-type methods for general variational inequalities. *Journal of Mathematical Analysis and Applications*, 277(2):379394, 2003. 7

[29] D. H. Peaceman and H. H. Rachford. The numerical solution of parabolic elliptic differential equations. *SIAM Journal on Applied Mathematics*, 3:28–41, 1955. 2

[30] K. Scheinberg, S. Ma, and D. Goldfarb. Sparse inverse covariance selection via alternating linearization methods. In *NIPS*, 2010. 2, 3

[31] 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. 7

[32] 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. 2

[33] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal Royal Statistical Society B*, 58:267–288, 1996. 13

[34] 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. 13

[35] 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. 2

[36] Z. Wen, D. Goldfarb, and W. Yin. Alternating direction augmented Lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, 2:203–230, 2010. 2

[37] J. Yang and X. Yuan. Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of Computation*, 82(281):301–329, 2013. 3

[38] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1 problems in compressive sensing. *SIAM Journal on Scientific Computing*, 33(1):250–278, 2011. 2, 3

[39] G. Ye and X. Xie. Split Bregman method for large scale fused Lasso. *Computational Statistics and Data Analysis*, 55(4):1552–1569, 2011. 13, 19

[40] M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007. 3

[41] X. Yuan. Alternating direction methods for sparse covariance selection. *Journal of Scientific Computing*, 51:261–273, 2012. 2, 3

[42] 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. 3

Table 1: Numerical Results for Basis Pursuit Problem

	GL			GAL			EGL			EGAL		
Inst.	Iter	$\|x-\hat{x}\|$	T	Iter	$\|x-\hat{x}\|$	T	Iter	$\|x-\hat{x}\|$	T	Iter	$\|x-\hat{x}\|$	T
$(n,m,s)=(100,20,2)$												
1	20000	6.1e-2	1.2	4211	1.7e-4	0.3	5020	1.9e-4	0.5	3171	1.7e-4	0.3
2	20000	2.9e-2	1.1	4100	4.3e-4	0.2	4283	5.9e-4	0.4	2457	3.4e-4	0.2
3	20000	1.4e-1	1.0	4411	1.2e-4	0.2	4407	2.5e-4	0.4	2216	1.5e-4	0.2
4	20000	5.7e-3	1.0	3278	2.8e-4	0.2	3497	2.7e-4	0.3	1353	3.3e-4	0.1
5	20000	3.7e-2	1.1	4402	6.1e-5	0.3	4399	1.9e-4	0.4	2345	6.4e-5	0.2
6	20000	1.1e-1	1.0	3914	3.0e-4	0.2	5561	2.2e-4	0.5	3084	2.0e-4	0.3
7	20000	4.5e-2	1.0	2859	2.0e-4	0.2	3721	1.6e-4	0.3	2389	1.9e-4	0.2
8	20000	2.0e-1	1.1	3977	1.6e-4	0.2	2760	1.6e-4	0.3	1889	1.2e-4	0.2
9	20000	7.9e-2	1.1	4612	1.9e-4	0.2	4609	8.4e-5	0.4	2208	3.3e-4	0.2
10	20000	8.3e-2	1.0	6431	2.1e-4	0.3	6428	6.0e-5	0.6	3441	1.1e-4	0.3
$(n,m,s)=(500,100,5)$												
1	20000	1.2e-1	17.1	6143	1.2e-4	5.3	6791	1.4e-4	11.9	3676	1.3e-4	6.8
2	20000	1.1e-1	18.0	6246	1.9e-4	5.5	6242	1.8e-4	10.5	3343	1.9e-4	5.6
3	20000	1.3e-1	18.2	5543	1.4e-4	5.3	5537	1.5e-4	9.6	3855	1.7e-4	6.6
4	20000	1.5e-2	18.2	7318	3.7e-4	6.6	7316	3.7e-4	12.5	3882	4.3e-4	6.7
5	20000	3.4e-2	17.8	6867	1.6e-4	6.4	7270	1.5e-4	12.6	3529	1.5e-4	5.9
6	20000	6.9e-2	17.8	6732	1.5e-4	6.5	6652	1.4e-4	11.6	3284	1.3e-4	5.7
7	20000	2.4e-3	17.6	7110	2.4e-4	6.3	7106	2.3e-4	12.3	5968	2.6e-4	10.3
8	20000	2.1e-1	17.7	8079	1.2e-4	7.3	8073	1.4e-4	14.0	4027	1.4e-4	7.1
9	20000	1.6e-1	17.9	7740	3.0e-4	6.9	7807	2.9e-4	13.4	3178	2.6e-4	5.6
10	20000	8.2e-4	22.3	8290	2.3e-4	8.1	8286	2.3e-4	14.7	5242	2.2e-4	9.5
$(n,m,s)=(1000,200,10)$												
1	20000	3.3e-2	79.6	9069	1.3e-4	36.9	8648	1.2e-4	70.9	5366	1.1e-4	44.9
2	20000	1.9e-1	95.0	8535	1.3e-4	40.3	8530	1.3e-4	84.1	4760	1.4e-4	40.5
3	20000	9.8e-2	87.0	8317	1.6e-4	35.9	8311	1.7e-4	71.2	4349	1.8e-4	37.6
4	20000	2.2e-2	87.2	8408	1.1e-4	45.3	8404	1.1e-4	79.5	5065	1.2e-4	48.1
5	20000	8.3e-2	96.8	7557	1.1e-4	33.5	7552	1.1e-4	66.4	4762	1.3e-4	41.7
6	20000	2.6e-2	89.3	9195	1.1e-4	41.6	9427	1.1e-4	83.1	4828	1.0e-4	42.6
7	20000	3.1e-2	89.3	9608	1.1e-4	42.6	9604	1.2e-4	83.7	7848	1.1e-4	69.1
8	20000	1.8e-2	88.1	9918	1.2e-4	44.4	9914	1.2e-4	87.5	8489	1.2e-4	73.8
9	20000	2.7e-4	90.1	13885	1.2e-4	61.7	13873	1.1e-4	122.2	12701	1.1e-4	110.1
10	20000	1.1e-1	86.4	8304	1.1e-4	35.9	8297	1.1e-4	70.9	4517	1.1e-4	38.9

Table 2: 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