Alternating Direction Methods of Multipliers with the BFGS update for Convex Optimization Problems

Yan Gu and Nobuo Yamashita

September 28, 2018

Abstract

Alternating direction methods of multipliers (ADMM) have been well studied and effectively used in various application fields. At each iteration, the classical ADMM must solve two subproblems exactly. To overcome the difficulty of computing the exact solution of the subproblems, some proximal terms are added to the subproblems. Recently, Gu and Yamashita studied a special proximal ADMM whose regularized matrix in the proximal term is generated by the BFGS update (or Limited memory BFGS) at every iteration for a structured quadratic optimization problem, and reported that the numbers of iterations were almost same as those by the exact ADMM in their numerical experiments. In this paper, we propose such a proximal ADMM with BFGS update for more general convex optimization problems. The convergence of the proposed method is proved under standard assumptions.

Keywords: alternating direction method, variable metric semi-proximal method, global convergence, BFGS method, limited memory BFGS, convex optimization.

1 Introduction

Consider the following convex optimization problem:

\[
\begin{align*}
\text{minimize} & \quad f(x) + g(y) \\
\text{subject to} & \quad Ax + By = b, \quad x \in \mathbb{R}^{n_1}, \quad y \in \mathbb{R}^{n_2},
\end{align*}
\] (1.1)

where \(f: \mathbb{R}^{n_1} \to \mathbb{R} \cup \{\infty\}\) and \(g: \mathbb{R}^{n_2} \to \mathbb{R} \cup \{\infty\}\) are proper convex functions, \(A \in \mathbb{R}^{m \times n_1}, B \in \mathbb{R}^{m \times n_2}\) and \(b \in \mathbb{R}^m\).

For solving (1.1) Gabay and Mercier [6], and Glowinski and Marrocco [7] proposed the alternating direction method of multipliers (ADMM) which generates the iterative sequence via the following recursion:

\[
\begin{align*}
x^{k+1} &= \arg \min_x L_\beta(x, y^k, \lambda^k), \\
y^{k+1} &= \arg \min_y L_\beta(x^{k+1}, y, \lambda^k), \\
\lambda^{k+1} &= \lambda^k - \beta(Ax^{k+1} - y^{k+1}),
\end{align*}
\] (1.2)

where \(L_\beta: \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^m \to \mathbb{R}\) is the augmented Lagrangian function for (1.1) defined by

\[
L_\beta(x, y, \lambda) := f(x) + g(y) - \langle \lambda, Ax + By \rangle + \frac{\beta}{2} \|Ax + By\|^2.
\] (1.3)

Here \(\lambda \in \mathbb{R}^m\) is multipliers associated to the equality constraints and \(\beta > 0\) is a penalty parameter.

The global convergence of the ADMM (1.2a)-(1.2c) can be established under very mild conditions [2].
Since the subproblems in (1.2a)-(1.2c) may be difficult to solve exactly in many applications, Eckstein [4] and He et al. [10] have considered to add proximal terms to the subproblems. Recently, Fazel et al. [5] proposed the following semi-proximal ADMM scheme:

\[
\begin{align*}
    x^{k+1} &= \arg \min_x \mathcal{L}_\beta(x, y^k, \lambda^k) + \frac{1}{2}\|x - x^k\|_T^2, \\
y^{k+1} &= \arg \min_y \mathcal{L}_\beta(x^{k+1}, y, \lambda^k) + \frac{1}{2}\|y - y^k\|_S^2, \\
    \lambda^{k+1} &= \lambda^k - \gamma \beta (Ax^{k+1} - y^{k+1}),
\end{align*}
\]

(1.4c)

where \( \gamma \in (0, (1 + \sqrt{5})/2) \), and \( \|z\|_G = \sqrt{z^T G z} \) for \( z \in \mathbb{R}^n \) and \( G \in \mathbb{R}^{n \times n} \). Fazel et al. [5] showed its global convergence when \( T \) and \( S \) are positive semidefinite, in contrast to the positive definite requirements in the classical proximal ADMM [4, 10], which makes the algorithm more flexible. See [3, 5, 11, 14] for the details.

In this paper, we allow the positive semidefinite matrix \( T \) in proximal term to be changed at every step. Then \( T \) depends on \( k \), and thus we denote it by \( T_k \). The resulting ADMM is a variable metric semi-proximal ADMM (VMSP-ADMM) algorithm given as

\[
\begin{align*}
    x^{k+1} &= \arg \min_x \mathcal{L}_\beta(x, y^k, \lambda^k) + \frac{1}{2}\|x - x^k\|_{T_k}^2, \\
y^{k+1} &= \arg \min_y \mathcal{L}_\beta(x^{k+1}, y, \lambda^k) + \frac{1}{2}\|y - y^k\|_S^2, \\
    \lambda^{k+1} &= \lambda^k - \beta (Ax^{k+1} - y^{k+1}),
\end{align*}
\]

(1.5c)

which is also related to the methods studied in He et al. [10] and Gonçalves et al. [8]. He et al. [10] assumed that the \( T_k \) was positive definite, while Gonçalves et al. [8] assumed that the \( T_k \) was positive semidefinite and proved the ergodic convergence rate of the variable metric proximal ADMM as a special case of a variable metric hybrid proximal extragradient framework for solving monotone inclusions.

For a special case that \( f(x) = \frac{1}{2}\|x\|^2, B = I \) and \( b = 0 \), Gu and Yamashita [9] proposed to construct the proximal term \( T_k \) as \( T_k = B_k - M \), where \( M = \nabla^2_{xx} \mathcal{L}_\beta(x, y, \lambda) = \beta A^T A + I \) and \( B_k \) is a certain positive definite matrix. Note that \( M \succ 0 \), where \( V \succeq 0 \) (\( V \succeq 0 \)) means \( V \) is symmetric and positive semidefinite (positive definite). Gu and Yamashita proposed to generate \( B_k \) via BFGS update with respect to \( M \) at every iteration. Then they showed that \( B_{k+1} \succeq M \) for all \( k \) whenever \( B_k \succeq M \). Therefore \( T_k \succeq 0 \) if the initial matrix \( B_0 \) satisfies \( B_0 \succeq M \). The variable metric semi-proximal ADMM with BFGS update (ADM-BFGS) for this special problem was given as

\[
\begin{align*}
    x^{k+1} &= x^k + H_k \left( A^T \lambda^k + \beta A^T y^k - M x^k \right), \\
y^{k+1} &= \arg \min_y \left\{ g(y) - \langle \lambda^k, A x^{k+1} - y \rangle + \frac{\beta}{2} \| A x^{k+1} - y \|^2 + \frac{1}{2} \| y - y^k \|_S^2 \right\}, \\
    \lambda^{k+1} &= \lambda^k - \beta (A x^{k+1} - y^{k+1}),
\end{align*}
\]

(1.6c)

where \( H_k = B_k^{-1} \). Gu and Yamashita reported that the numbers of iterations were almost same as those by the exact ADMM in their numerical experiments.

In this paper, we extend their method to consider more general problems. In particulars, we consider the following two problems.

One is formulated as

\[
\text{Problem 1:} \quad \text{minimize} \quad \sum_{i=1}^N f_i(A_i x) \quad \text{subject to} \quad x \in \mathbb{R}^n,
\]

(1.7)

where \( f_i : \mathbb{R}^{m_i} \to \mathbb{R} \cup \{ \infty \} \) is a proper convex function, and \( A_i \in \mathbb{R}^{m_i \times n} \) for all \( i = 1, 2, \ldots, N \).
The other problem is

\begin{align}
\text{Problem 2:} \quad & \text{minimize} \quad \sum_{i=1}^{N} f_i(x_i) \\
& \text{subject to} \quad \sum_{i=1}^{N} A_i x_i = b, \quad x_i \in \mathbb{R}^{n_i}, \quad i = 1, 2, \ldots, N
\end{align}

where \( f_i : \mathbb{R}^{n_i} \to \mathbb{R} \cup \{ \infty \} \) is a proper convex function, \( A_i \in \mathbb{R}^{m \times n_i} \) for all \( i = 1, 2, \ldots, N \) and \( b \in \mathbb{R}^m \).

Now, we apply the ADMM (1.5a)-(1.5b) to the above two convex problems. To this end, we reformulate the problems as (1.1). By introducing some auxiliary variables \( y_i \in \mathbb{R}^m (i = 1, 2, \ldots, N) \), problem (1.7) can be reformulated as

\begin{align}
\text{minimize} \quad & \sum_{i=1}^{N} f_i(y_i) \\
& \text{subject to} \quad y_i = A_i x_i, \quad i = 1, 2, \ldots, N \\
& \quad x \in \mathbb{R}^n, y_i \in \mathbb{R}^m, \quad i = 1, 2, \ldots, N.
\end{align}

Letting \( y = (y_1^T, y_2^T, \ldots, y_N^T)^T, \quad f(x) \equiv 0, \quad g(y) = \sum_{i=1}^{N} f_i(y_i), \quad A = [A_1^T, A_2^T, \ldots, A_N^T]^T, \quad B = -I, \quad b = 0 \), problem (1.9) is reduced to (1.1).

Similarly, by introducing some auxiliary variables \( y_i \in \mathbb{R}^{n_i} (i = 1, 2, \ldots, N) \), problem (1.8) can be reformulated as

\begin{align}
\text{minimize} \quad & \sum_{i=1}^{N} f_i(y_i) \\
& \text{subject to} \quad \sum_{i=1}^{N} A_i x_i = b, \\
& \quad x_i = y_i, \quad i = 1, 2, \ldots, N \\
& \quad x_i, y_i \in \mathbb{R}^{n_i}, \quad i = 1, 2, \ldots, N.
\end{align}

Letting \( x = (x_1^T, x_2^T, \ldots, x_N^T)^T, \quad y = (y_1^T, y_2^T, \ldots, y_N^T)^T, \quad f(x) \equiv 0, \quad g(y) = \sum_{i=1}^{N} f_i(y_i), \quad \bar{A} = [A_1, A_2, \ldots, A_N], \)

and \( A = \begin{bmatrix} \bar{A} \\ I \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ -I \end{bmatrix}, \quad b = 0 \), problem (1.10) is also reduced to (1.1).

The main contributions of the paper are as follows:

1. We give sufficient conditions on positive semidefinite \( T_k \) under which a sequence \( \{ x^k \} \) generated by VMSP-ADMM (1.5) for the above two problems globally converges to their solutions.

2. We propose update formulae of \( T_k \) for the general problems with BFGS which satisfy the above sufficient conditions for global convergence.

The rest of the paper is organized as follows. We describe the construction of \( T_k \) via the BFGS update and show the details on applying the ADMM for the above two convex problems (1.7) and (1.8) in Section 2. Some preliminary convergence properties of the general variable metric semi-proximal ADMM are provided in Section 3. In Section 3, we also discuss the convergence of the proposed method under certain flexible conditions on the proximal matrices sequence. In Section 4, we test L1 regularized logistic regression problem to illustrate the efficiency of the proposed method. Finally, we make some concluding remarks in Section 5.

## 2 ADMMs for two convex optimization problems

In this section, we first explain how to construct \( T_k \) via BFGS update, and present concrete algorithms for two convex optimization problems (1.7) and (1.8).
2.1 Construction of the regularized matrix $T_k$ via the BFGS update

Throughout this paper, we suppose that $x$-subproblems (1.5a) are unconstrained quadratic programming problem, and that the Hessian matrix of the augmented Lagrangian function (1.3) is a constant matrix defined as

$$M := \nabla^2_{xx} \mathcal{L}_\beta(x, y, \lambda).$$

At first, we should give the following assumption.

**Assumption 2.1.** $M$ is positive semidefinite.

The Hessian of the objective function of $x$-subproblem (1.5a) is $B = T_k + M$. Note that if $T_k = 0$, that is, we consider standard ADMM, then $B = M$. In order to avoid computing the inverse of $M$ but still has some information on $M$, we consider a matrix $B$ that has the following three properties:

(i) $T = B - M$;

(ii) $B \succeq M$;

(iii) $B$ has some second order information on $M$.

We want $T$ to be positive semidefinite for global convergence and $B$ to be as close to $M$ as possible for rapid convergence. To this end, Gu and Yamashita [9] proposed to construct $B$ via the BFGS update with respect to $M$ at every iteration. The proximal term $B$ and $T$ depend on $k$ and become as $B_k$ and $T_k$, that is, $T_k = B_k - M$.

Since BFGS usually constructs the inverse of $B_k$, let $H_k = B_k^{-1}$. Using $H_k$, we can easily solve the $x$-subproblems.

When $M > 0$, we consider the normal BFGS update with a given $s \in \mathbb{R}^n$ and $l = Ms$. Note that $s^\top l > 0$ when $s \neq 0$. Then BFGS recursion for $B_{next}$ and $H_{next}$ are given as

$$B_{next} = B + \frac{ll^\top}{l^\top s} - \frac{Bss^\top B^\top}{s^\top Bs},$$

$$H_{next} = \left(I - \frac{s^\top}{s^\top l} l\right) H \left( I - \frac{l s^\top}{s^\top l} l\right) + \frac{ss^\top}{s^\top l},$$

where $B$ and $H$ are the proximal matrices for the current step, $B_{next}$ and $H_{next}$ are the new matrices generated via BFGS update with $s$ and $l$ for the next iteration.

Since $s^\top l > 0$, $B_{next}$ and $H_{next}$ are positive definite whenever $B, H > 0$. Moreover Gu and Yamashita [9] showed the following useful property.

**Theorem 2.2.** [9, Theorem 3.2] Let $s \in \mathbb{R}^n$ such that $s \neq 0$, and let $l = Ms$. If $H \preceq M^{-1}$, then $H_{next} \preceq M^{-1}$.

Due to this theorem we show that the $T_k$ is positive semidefinite when $B_k$ is updated by BFGS method for the global convergence. That is, $T_k = B_k - M \succeq 0$ when the initial matrix $H_0$ satisfies

$$H_0 \preceq M^{-1},$$

where $H_k$ is the matrix generated from BFGS (2.1) and $B_k = (H_k)^{-1}$.

When $M \succeq 0$, we cannot directly use BFGS update. In this case we may consider $M_{\delta} := M + \delta I$ with sufficiently small $\delta > 0$, and construct $B_k$ by BFGS update for $M_{\delta}$, that is, $B$ is updated by BFGS with respect to $M_{\delta}$ at every iteration and $\ell = M_{\delta} s = Ms + \delta s$. Then $T_k = B_k - M \succeq \delta I > 0$ is a positive definite matrix. Note that $s^\top \ell > 0$ when $s \neq 0$ and the BFGS recursion (2.1) still holds. When the $\delta$ is small enough, $B$ is also close to $M$. 

2.2 VMSP-ADMM for convex problem 1

In this subsection, we describe the details of applying the VMSP-ADMM with BFGS update for the two convex problems (1.7) and (1.8) shown in Introduction.

First we consider the convex problem 1 (1.7). Let \( L^\beta_b(x, y_1, ..., y_N, \lambda_1, ..., \lambda_N) \) be the augmented Lagrangian function for (1.9) that defined by

\[
L^\beta_b(x, y_1, ..., y_N, \lambda_1, ..., \lambda_N) := \sum_{i=1}^{N} f_i(y_i) - \sum_{i=1}^{N} \langle \lambda_i, A_i x - y_i \rangle + \sum_{i=1}^{N} \frac{\beta}{2} \| A_i x - y_i \|^2, \tag{2.2}
\]

where \( \lambda_i \in \mathbb{R}^{m_i} (i = 1, 2, ..., N) \) are multipliers associated to the linear constraints and \( \beta > 0 \) is a penalty parameter.

Besides, there is no coupling relationship for all the \( y_i, i = 1, 2, ..., N \). Similar as that in Introduction, we use

\[
g(y) = \sum_{i=1}^{N} f_i(y_i), \quad y = (y_1^T, y_2^T, ..., y_N^T)^T, \quad \text{and} \quad A_1 = [A_1^T, A_2^T, ..., A_N^T]^T, \quad m = \sum_{i=1}^{N} m_i, \quad \lambda = (\lambda_1^T, \lambda_2^T, ..., \lambda_N^T)^T,
\]

then \( y \in \mathbb{R}^m \) and \( \lambda \in \mathbb{R}^m \). Then the augmented Lagrangian function is rewritten as

\[
L^\beta_b(x, y, \lambda) := g(y) - \langle \lambda, A_1 x - y \rangle + \frac{\beta}{2} \| A_1 x - y \|^2. \tag{2.3}
\]

The \( x \)-subproblem (1.5a) for this problem is written as

\[
x^{k+1} = \arg \min_{x} \left\{ -\langle \lambda^k, A_1 x - y^k \rangle + \frac{\beta}{2} \| A_1 x - y^k \|^2 + \frac{1}{2} \| x - x^k \|^2_T \right\}
\]

\[
= \left( \beta A_1^T A_1 + T_k^1 \right)^{-1} \left( A_1^\top \lambda^k + \beta A_1^\top y^k + T_k^1 x^k \right).
\]

Note that \( \beta A_1^T A_1 = \sum_{i=1}^{N} \beta A_i^T A_i = \nabla^2_{xx} L^\beta_b(x, y, \lambda) \). Let \( M^1 \) be defined as the Hessian matrix of the augmented Lagrangian function \( L^\beta_b \), that is,

\[
M^1 := \nabla^2_{xx} L^\beta_b(x, y, \lambda) = \beta A_1^T A_1 \succeq 0.
\]

Note that \( M^1 \) is not necessarily positive definite. Then, as written in the end of the previous subsection, we may not apply BFGS update for \( M^1 \). Therefore we use the following perturbed matrix \( M^{\delta_1} \) instead of \( M^1 \).

\[
M^{\delta_1} := M^1 + \delta_1 I \succ 0 \quad \text{with} \quad \delta_1 > 0.
\]

Then, we consider to construct a matrix \( B^1_k \) that satisfies:

\[
T_k^1 = B^1_k - M^1 \quad \text{with} \quad B^1_k \succeq M^{\delta_1} \succ M^1.
\]

Using \( B^1_k \), \( x \)-subproblem (1.5a) is written as

\[
x^{k+1} = \arg \min_{x} \left\{ -\langle \lambda^k, x - y^k \rangle + \frac{\beta}{2} \| x - y^k \|^2 + \frac{1}{2} \| x - x^k \|^2_T \right\}
\]

\[
= x^k + (B^1_k)^{-1} (A_1^\top \lambda^k + \beta A_1^\top y^k - M^1 x^k). \tag{2.4}
\]

We propose to construct \( B^1_k \) via the BFGS update with respect to \( M^{\delta_1} \) at every iteration.

Based on the description of the BFGS (or Limited memory BFGS, abbreviated to L-BFGS) update, we first give our algorithm for the above problem in detail.

**Remark 2.3.** Note that the constant \( \bar{k} \in [1, \infty] \) in the algorithm means that the \( B^1_{\bar{k}} (H^1_{\bar{k}}) \) updated by the BFGS (or L-BFGS) procedure will be stopped at \( \bar{k} \), that is, \( B^1_{\bar{k}} = B^1_{\bar{k}+1}, \quad (H^1_{\bar{k}} = H^1_{\bar{k}+1}) \) for \( k \geq \bar{k} \). Specially,

- \( \bar{k} < \infty \) : we can show the global convergence under this condition;
- \( \bar{k} = \infty \) : the \( B^1_{\bar{k}} (H^1_{\bar{k}}) \) are updated for all \( k \), and the numerical experiments in [9] show it works;
- \( \bar{k} \) is small: it is not efficient, since \( B^1_{\bar{k}} \) is not close to \( M^{\delta_1} \).
Algorithm 1: VMSP-ADMM with the BFGS update (ADM-B1) for the convex Problem 1 (1.7)

\textbf{Input}: size \((m, n)\), data matrix \(A_1\), initial point \((x^0, y^0, \lambda^0)\), penalty parameter \(\beta, \delta_1, \text{maxIter}\);
initial matrix \(H_0^1 \preceq (M^\delta_1)^{-1}\), constant \(\tilde{k}^1 \in [1, \infty]\), stopping criterion \(\epsilon\).

\textbf{Output}: approximative solution \((x^k, y^k, \lambda^k)\)

1 initialization;
2 \textbf{while} \(k < \text{maxIter}\) or not convergence \textbf{do}
3 \hspace{1em} \textbf{if} \(k \leq \tilde{k}^1\) and \(x_k - x_{k-1} \neq 0\) \textbf{then}
4 \hspace{2em} \textbf{update} \(H^1_k\) via BFGS (or L-BFGS) with the initial matrix \(H^1_0\);
5 \hspace{1em} \textbf{else}
6 \hspace{2em} \(H^1_k = H^1_{k-1}\);
7 \textbf{end}
8 \textbf{update} the \(x^{k+1}\) by solving the \(x\)-subproblem: \(x^{k+1} = x^k + H^1_k (A^T_1 \lambda^k + \beta A^T_1 y^k - M^1 x^k)\);
9 \textbf{update} the \(y^{k+1}\) by solving the \(y\)-subproblem:
10 \hspace{1em} \(y^{k+1} = \arg \min_y \left\{ g(y) - (\lambda^k, A_1 x^{k+1} - y) + \frac{\beta}{2} \| A_1 x^{k+1} - y \|^2 + \frac{1}{2} \| y - y^k \|^2 \right\}\);
11 \textbf{update} the augmented lagrangian parameter: \(\lambda^{k+1} = \lambda^k - \beta (A_1 x^{k+1} - y^{k+1})\).
12 \textbf{end}

2.3 VMSP-ADMM for convex problem 2

For the convex problem 2 (1.8), let \(L^2_\beta(x_1, \ldots, x_N, y_1, \ldots, y_N, \lambda, \mu_1, \ldots, \mu_N)\) be the augmented Lagrangian function for (1.10) that defined by

\[
L^2_\beta(x_1, \ldots, x_N, y_1, \ldots, y_N, \lambda, \mu_1, \ldots, \mu_N) := \sum_{i=1}^N f_i(y_i) - \langle \lambda, \sum_{i=1}^N A_i x_i - b \rangle - \sum_{i=1}^N \langle \mu_i, x_i - y_i \rangle + \frac{\beta_1}{2} \| \sum_{i=1}^N A_i x_i - b \|^2 + \sum_{i=1}^N \frac{\beta_2}{2} \| x_i - y_i \|^2, \quad (2.5)
\]

where \(\lambda \in \mathbb{R}^m, \mu_i \in \mathbb{R}^n_i (i = 1, 2, \ldots, N)\) are multipliers associated to the linear constraints and \(\beta_1, \beta_2 > 0\) are the penalty parameters, respectively.

We also define \(x = (x_1^T, x_2^T, \ldots, x_N^T)^T, y = (y_1^T, y_2^T, \ldots, y_N^T)^T, A_2 = [A_1, A_2, \ldots, A_N], n = \sum_{i=1}^N n_i\), then \(x, y \in \mathbb{R}^n\) and \(A_2 \in \mathbb{R}^{m \times n}\). Moreover \(g(y) = \sum_{i=1}^N f_i(y_i), \mu = (\mu_1^T, \mu_2^T, \ldots, \mu_N^T)^T \) with \(\mu \in \mathbb{R}^n\). The augmented Lagrangian function (2.5) can be written as

\[
L^2_\beta(x, y, \lambda, \mu) = g(y) - \langle \lambda, A_2 x - b \rangle - \langle \mu, x - y \rangle + \frac{\beta_1}{2} \| A_2 x - b \|^2 + \frac{\beta_2}{2} \| x - y \|^2. \quad (2.6)
\]

As the same in subsection 2.2.1, the \(x\)-subproblem (1.5a) for this problem 2 is written as

\[
x^{k+1} = \arg \min_x \left\{ -\langle \lambda^k, A_2 x - b \rangle - \langle \mu^k, x - y^k \rangle + \frac{\beta_1}{2} \| A_2 x - b \|^2 + \frac{\beta_2}{2} \| x - y^k \|^2 + \frac{1}{2} \| x - x^k \|^2 \right\}
\]

\[
= \left( \beta_1 A_2^T A_2 + \beta_2 I + T^k_2 \right)^{-1} \left( \beta_1 A_2^T \lambda^k + \mu^k + \beta_1 A_2^T b + \beta_2 y^k + T^k_2 x^k \right).
\]

Let \(M^2\) be defined as the Hessian matrix of the augmented Lagrangian function \(L^2_\beta\), that is,

\[
M^2 := \nabla^2_{xx} L^2_\beta(x, y, \lambda, \mu) = \beta_1 A_2^T A_2 + \beta_2 I.
\]

\(M^2 > 0\) whenever \(\beta_2 > 0\). Then, we consider to construct a matrix \(B^2_k\) that satisfies:

\[
T^k_2 = B^2_k - M^2 \text{ with } B^2_k \succeq M^2.
\]
Using $B_k^2$, $x$-subproblem (1.4a) is written as

$$
x^{k+1} = \arg \min_x \left\{ -(\lambda^k, A_x x - b) - (\mu^k, x - y^k) + \frac{\beta_1}{2} \norm{A_x x - b}^2 + \frac{\beta_2}{2} \norm{x - y^k}^2 + \frac{1}{2} \norm{x - x^k}_T^2 \right\}
$$

$$
x^{k+1} = x^k + (B_k^2)^{-1} (A_x^T \lambda^k + \mu^k + \beta_1 A_x^T b + \beta_2 y^k - M^2 x^k)
$$

(2.7)

We construct $B_k^2$ via the BFGS update at every iteration. The algorithm for the above problem 2 is:

**Algorithm 2: VMSP-ADMM with the BFGS update (ADM-B2) for the convex Problem 2 (1.8)**

1. initialization;
2. while $k < \maxIter$ or not convergence do
3.    if $k \leq \bar{k}$ and $x_k - x_{k-1} \neq 0$ then
4.        update $H_k^2$ via BFGS (or L-BFGS) with the initial matrix $H_0^2$;
5.    else
6.        $H_k^2 = H_{k-1}^2$;
7.    end
8.    update the $x^{k+1}$ by solving the $x-$subproblem:
9.        $x^{k+1} = x^k + H_k^2 (A_x^T \lambda^k + \mu^k + \beta_1 A_x^T b + \beta_2 y^k - M^2 x^k)$;
10. update the $y^{k+1}$ by solving the $y-$subproblem:
11.        $y^{k+1} = \arg \min_y \left\{ g(y) - (\mu^k, x^{k+1} - y) + \frac{\beta_2}{2} \norm{x^{k+1} - y}^2 + \frac{1}{2} \norm{y - y^k}_{S^2} \right\}$;
12. update the augmented lagrangian parameter: $\lambda^{k+1} = \lambda^k - \beta_1 (A_x x^{k+1} - b)$;
13. update the augmented lagrangian parameter: $\mu^{k+1} = \mu^k - \beta_2 (x^{k+1} - y^{k+1})$.
14. end

The $\bar{k}$ in the Algorithm 2 plays the same role as the $\bar{k}$ in Algorithm 1, see from Remark 2.3.

## 3 Convergence of VMSP-ADMM for the Convex problem

In this section, we first consider the variable metric semi-proximal ADMM (1.5) for problem (1.1) with convex functions $f$ and $g$. We give the optimality condition of problem (1.1) and some properties which will be frequently used in our analysis. Then we show general convergence properties for the variable metric semi-proximal ADMM (1.5a)-(1.5c) (VMSP-ADMM). At last we derive the convergence of VMSP-ADMM with BFGS update (ADM-B1 and ADM-B2) for problem 1 (1.7) and problem 2 (1.8).

### 3.1 Preliminaries

The KKT conditions of problem (1.1) are written as:

\[
\begin{align*}
  f'(x^*) - A^T \lambda^* &= 0, \\
  g'(y^*) - B^T \lambda^* &= 0, \\
  Ax^* + By^* - b &= 0,
\end{align*}
\]

where $f'(x^*) \in \partial f(x^*)$ and $g'(y^*) \in \partial g(y^*)$.

Let $\Omega^*$ be a set of $(x^*, y^*, \lambda^*)$ satisfying the KKT condition (3.1). Throughout this paper, we make the following assumptions.
Assumption 3.1. The set $\Omega^*$ of KKT points is non-empty.

Let
\[
F(w) = \begin{pmatrix} f'(x) - A^T \lambda \\ g'(y) - B^T \lambda \\ Ax + By - b \end{pmatrix},
\]
where $f'(x) \in \partial f(x)$ and $g'(y) \in \partial g(y)$, and let $\Omega = \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m$.

Since the subdifferential mapping of the closed proper convex functions are maximal monotone [13], there exist two positive semidefinite matrices $\Sigma_f$ and $\Sigma_g$ such that for all $x, \hat{x} \in \mathbb{R}^n$, $f'(x) \in \partial f(x)$, and $f'(\hat{x}) \in \partial f(\hat{x})$,
\[
(x - \hat{x})^T (f(x) - f(\hat{x})) \geq \|x - \hat{x}\|_{\Sigma_f}^2,
\]
and for all $y, \hat{y} \in \mathbb{R}^m$, $g'(y) \in \partial g(y)$, and $g'(\hat{y}) \in \partial g(\hat{y})$,
\[
(y - \hat{y})^T (g(y) - g(\hat{y})) \geq \|y - \hat{y}\|_{\Sigma_g}^2.
\]

We make the following assumption on $T_k$ and $S$ for global convergence.

Assumption 3.2. For all $k > 0$, $T_k + \Sigma_f + \beta A^T A$ and $S + \Sigma_g + \beta B^T B$ are positive definite.

In Subsection 3.3, we will show that the Assumption 3.2 holds when $S$ is positive definite and $T_k$ is generated as in Subsection 2.1.

Let $(x^0, y^0, \lambda^0) \in \Omega$ be the initial arbitrary triplet. By deriving the first-order optimality conditions of the involved subproblems in (1.5), we can easily show that for the given triplet $(x^k, y^k, \lambda^k) \in \Omega$, and $\beta > 0$, the new triplet $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ is generated by the following procedure.

- **step 1:** Find $x^{k+1} \in \mathbb{R}^n$ such that $f'(x^{k+1}) \in \partial f(x^{k+1})$ and
\[
f'(x^{k+1}) - A^T \lambda^k + A^T \beta (Ax^{k+1} + By^k - b) + T_k (x^{k+1} - x^k) = 0,
\]

- **step 2:** Find $y^{k+1} \in \mathbb{R}^m$ such that $g'(y^{k+1}) \in \partial g(y^{k+1})$ and
\[
g'(y^{k+1}) - B^T \lambda^k + B^T \beta (Ax^{k+1} + By^{k+1} - b) + S (y^{k+1} - y^k) = 0,
\]

- **step 3:** Update $\lambda^{k+1}$ via
\[
\lambda^{k+1} = \lambda^k - \beta (Ax^{k+1} + By^{k+1} - b).
\]

In the following analysis, we will use (3.4) to (3.6).

### 3.2 Global Convergence of variable metric semi-proximal ADMM

In this subsection, we show VMSP-ADMM (3.4)-(3.6) converges to a solution of (1.1) globally. To this end, we first show that a sequence related to $\{(x^k, y^k, \lambda^k)\}$ generated by (1.5) is contractive.

For $k = 0, 1, 2, \ldots$, we use the following notation:
\[
u^* = \begin{pmatrix} x^* \\ y^* \end{pmatrix}, \quad u^k = \begin{pmatrix} x^k \\ y^k \end{pmatrix}, \quad w^k = \begin{pmatrix} x^k \\ y^k \\ \lambda^k \end{pmatrix}, \quad \text{and} \quad D_k = \begin{pmatrix} T_k & 0 \\ 0 & S \end{pmatrix}.
\]

Moreover, for simplicity, we denote
\[
F(w^k) = \begin{pmatrix} f'(x^k) - A^T \lambda^k \\ g'(y^k) - B^T \lambda^k \\ Ax^k + By^k - b \end{pmatrix},
\]
where $f'(x^k)$ and $g'(y^k)$ are obtained in steps 1 and 2 in VMSP-ADMM. Note that $F$ is a set to point map in general.
Lemma 3.3. Let \( \{w^k\} \) be generated by (1.5). Then, for given \( w^* = (x^*, y^*, \lambda^*) \in \Omega^* \), we have
\[
(u^{k+1} - u^*)^\top D_k(u^{k+1} - u^k) + \frac{1}{\beta}(\lambda^{k+1} - \lambda^*)^\top(\lambda^{k+1} - \lambda^k) + \|x^{k+1} - x^*\|_{\Sigma_x}^2 + \|y^{k+1} - y^*\|_{\Sigma_y}^2 \\
\leq \beta(Ax^{k+1} - Ax^*)^\top(By^{k+1} - By^*).
\] (3.8)

Proof. Since the subdifferential mappings \( \partial f \) and \( \partial g \) are maximal monotone and (3.2)-(3.3), we have
\[
(x^{k+1} - x^*)^\top(f'(x^{k+1}) - f'(x^*)) \geq \|x^{k+1} - x^*\|_{\Sigma_x}^2
\] (3.9)
and
\[
(y^{k+1} - y^*)^\top(g'(y^{k+1}) - g'(y^*)) \geq \|y^{k+1} - y^*\|_{\Sigma_y}^2.
\] (3.10)
Combined with KKT conditions (3.1) and (3.4)-(3.5), the assertion can be shown in a way similar to the proof of [9, Lemma 2.3].

Let \( \Sigma \in \mathbb{R}^{2n \times 2n} \) denote
\[
\Sigma = \left( \begin{array}{cc} \Sigma_f & 0 \\ 0 & \Sigma_g \end{array} \right).
\] (3.11)

Then \( \|x^{k+1} - x^*\|_{\Sigma_x}^2 + \|y^{k+1} - y^*\|_{\Sigma_y}^2 = \|u^{k+1} - u^*\|_{\Sigma}^2 \).

Lemma 3.4. Let \( w^* = (x^*, y^*, \lambda^*) \in \Omega^* \), and let \( \{w^k\} \) be generated by the scheme (1.5). Then it holds that
\[
\|u^k - u^*\|_{\Sigma_u}^2 + \frac{1}{\beta}\|\lambda^k - \lambda^*\|^2 + \beta\|B(y^k - y^*)\|^2 + \|y^k - y^{k-1}\|_{\Sigma_y}^2 \\
- \left(\|u^{k+1} - u^*\|_{\Sigma_u}^2 + \frac{1}{\beta}\|\lambda^{k+1} - \lambda^*\|^2 + \beta\|B(y^{k+1} - y^*)\|^2 + \|y^{k+1} - y^k\|_{\Sigma_y}^2\right) \\
\geq \|u^{k+1} - u^k\|_{\Sigma_u}^2 + \beta\|B(y^{k+1} - y^k)\|^2 + \beta\|Ax^{k+1} + By^{k+1} - (\lambda^k)\|^2 + 2\|u^{k+1} - u^*\|_{\Sigma}^2.
\] (3.12)

Proof. Since \( Ax^* + By^* = b \), the right hand term \( \beta(Ax^{k+1} - Ax^*)^\top(By^{k+1} - By^*) \) in (3.8) of Lemma 3.3 is
\[
\beta(Ax^{k+1} - Ax^*)^\top(By^{k+1} - By^*) = \beta(B(y^{k+1} - y^*)^\top(Ax^{k+1} + By^{k+1} - b) - \beta(B(y^{k+1} - y^*))^\top(B(y^{k+1} - y^*)).
\]
We shall estimate the first term \( \beta(B(y^{k+1} - y^*))^\top(Ax^{k+1} + By^{k+1} - b) \).

From (3.4)-(3.5), we have
\[
B^\top \lambda^{k+1} - S(y^{k+1} - y^k) = g'(y^{k+1}), \quad \text{and} \quad B^\top \lambda^k - S(y^k - y^{k-1}) = g'(y^k),
\]
where \( g'(y^{k+1}) \in \partial g(y^{k+1}) \) and \( g'(y^k) \in \partial g(y^k) \). We obtain from the monotonicity of \( \partial g \) that
\[
(y^{k+1} - y^k)^\top(B^\top \lambda^{k+1} - S(y^{k+1} - y^k) - (B^\top \lambda^k - S(y^k - y^{k-1}))) \geq 0,
\]
and hence
\[
(B(y^{k+1} - y^k))^\top(\lambda^{k+1} - \lambda^k) \geq \|y^{k+1} - y^k\|_{\Sigma_y}^2 - (y^{k+1} - y^k)^\top S(y^k - y^{k-1}).
\]
It then follows from \( \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b) \) that
\[
\beta(B(y^{k+1} - y^k))^\top(Ax^{k+1} + By^{k+1} - b) = (B(y^{k+1} - y^k))^\top(\lambda^{k+1} - \lambda^k) \\
\leq -\|y^{k+1} - y^k\|_{\Sigma_y}^2 + (y^{k+1} - y^k)^\top S(y^k - y^{k-1}) \\
\leq \frac{1}{2}\|y^k - y^{k-1}\|_{\Sigma_y}^2 - \frac{1}{2}\|y^{k+1} - y^k\|_{\Sigma_y}^2.
\] (3.13)

Using \( (u^{k+1} - u^*)^\top D_k(u^{k+1} - u^k) + \frac{1}{\beta}(\lambda^{k+1} - \lambda^*)^\top(\lambda^{k+1} - \lambda^k) + \|x^{k+1} - x^*\|_{\Sigma_x}^2 + \|y^{k+1} - y^*\|_{\Sigma_y}^2 \\
\leq -\beta(B(y^{k+1} - y^k))^\top(B(y^{k+1} - y^*)) + \frac{1}{2}\|y^k - y^{k-1}\|_{\Sigma_y}^2 - \frac{1}{2}\|y^{k+1} - y^k\|_{\Sigma_y}^2. \) (3.14)

Using the inequality (3.14) and usual analysis as in the proof of [9, Lemma 2.4], we get the assertion of this lemma.
Next we give an upper bound for the residual $\|F(w^{k+1})\|$.

**Lemma 3.5.** Let $w^* = (x^*, y^*, A^*) \in \Omega^*$, and let $\{u^k\}$ be generated by the scheme (1.5). Suppose that a sequence $\{T_k\}$ is bounded. Then, there exists a constant $\mu > 0$ such that for all $k \geq 0$, we have

$$\|F(w^{k+1})\| \leq \mu \left(\|u^{k+1} - u^k\|_{D_k}^2 + \|B(y^{k+1} - y^k)\|^2 + \|Ax^{k+1} + By^k - b\|^2\right). \quad (3.15)$$

**Proof.** From (3.7) and (3.4)-(3.6), we get

$$\|F(w^{k+1})\| = \left\| \begin{array}{c}
  f'(x^{k+1}) - A^T x^{k+1} \\
  g'(y^{k+1}) - B^T y^{k+1} \\
  Ax^{k+1} + By^{k+1} - b
\end{array} \right\|$$

$$= \left\| \begin{array}{c}
  \beta A^T B(y^{k+1} - y^k) - T_k(x^{k+1} - x^k) \\
  -S(y^{k+1} - y^k) \\
  Ax^{k+1} + By^{k+1} - b
\end{array} \right\|$$

$$\leq \|T_k\|\|x^{k+1} - x^k\| + \|S\|\|y^{k+1} - y^k\| + \beta\|A^T\|\|B(y^{k+1} - y^k)\| + \|Ax^{k+1} + By^{k+1} - b\|$$

$$\leq c_k\|x^{k+1} - x^k\| + \|y^{k+1} - y^k\| + \|B(y^{k+1} - y^k)\| + \|Ax^{k+1} + By^{k+1} - b\|,$$

where $c_k = \max\{\|T_k\|, \beta\|A^T\|, \|S\|, 1\}$. Since $\{T_k\}$ is bounded, then it follows from the above inequality that there exists a constant $\mu > 0$ such that

$$\|F(w^{k+1})\| \leq \mu \left(\|u^{k+1} - u^k\|_{D_k}^2 + \|B(y^{k+1} - y^k)\|^2 + \|Ax^{k+1} + By^{k+1} - b\|^2\right). \quad \square$$

Now, we begin to investigate the convergence of VMSP-ADMM. First, we assume some conditions for sequence $\{T_k\}$ (i.e., $\{D_k\}$) that should be obeyed to guarantee the convergence.

**Condition 3.1.** For the sequence $\{T_k\}$ generated by the framework (1.5), there exist $T \succeq 0$ and a non-negative sequence $\{\gamma_k\}$ such that

1. $T \preceq T_{k+1} \preceq (1 + \gamma_k)T_k$, $\forall k \geq 0$,
2. $T + \Sigma_f + \beta A^T A$ is positive definite,
3. $\sum_0^{\infty} \gamma_k < \infty$.

From the definition of $\{D_k\}$, it follows that the sequence $\{D_k\}$ also satisfy $D \preceq D_{k+1} \preceq (1 + \gamma_k)D_k$ for all $k$, where $D = \begin{pmatrix} T & 0 \\ 0 & S \end{pmatrix}$. We define two constants $C_s$ and $C_p$ as follows:

$$C_s := \sum_{k=0}^{\infty} \gamma_k \quad \text{and} \quad C_p := \prod_{k=0}^{\infty} (1 + \gamma_k). \quad \text{(3.16)}$$

From the assumption $\sum_0^{\infty} \gamma_k < \infty$ and $\gamma_k \geq 0$, it follows that $0 \leq C_s < \infty$ and $1 \leq C_p < \infty$. We can easily get

$$T \preceq T_k \preceq C_p T_0, \quad \forall k \geq 0,$$

which means that the sequences $\{T_k\}$ and $\{D_k\}$ are bounded.

For convenience, we denote the following matrix:

$$G_k = \begin{pmatrix} T_k + \Sigma_f & 0 & 0 \\ 0 & S + \Sigma_g + \beta B^T B & 0 \\ 0 & 0 & \frac{1}{\beta} I \end{pmatrix} \quad \text{and} \quad \bar{G} = \begin{pmatrix} T + \Sigma_f & 0 & 0 \\ 0 & S + \Sigma_g + \beta B^T B & 0 \\ 0 & 0 & \frac{1}{\beta} I \end{pmatrix}. \quad \text{(3.17)}$$

Obviously, $G_k \succeq \bar{G} \succeq 0$ since $T \succeq 0, S \succeq 0$ and $\beta > 0$.

Now we give the main theorem of this subsection.
Theorem 3.6. Let \( w^* = (x^*, y^*, \lambda^*) \in \Omega^* \), \( \{w^k\} \) be generated by (1.5), and \( \{T_k\} \) be a sequence satisfying Condition 3.1. Suppose that Assumptions 3.1 and 3.2 hold. Then the sequence \( \{w^k\} \) converges to a point \( w^* \in \Omega^* \).

Proof. First we show that the sequence \( \{w^k\} \) is bounded.

It is trivial to know from (3.12) in Lemma 3.4 that
\[
\lim_{k \to \infty} \left( ||u^{k+1} - u^k||^2_{D_k} + \beta ||B(y^{k+1} - y^k)||^2 + \beta ||Ax^{k+1} + By^{k+1} - b||^2 + 2||u^{k+1} - u^*||^2_{\Sigma_k} \right) = 0,
\]
which indicates that
\[
\lim_{k \to \infty} ||B(y^{k+1} - y^k)|| = \lim_{k \to \infty} ||Ax^{k+1} + By^{k+1} - b|| = 0.
\] (3.18)

Besides, it is straightforward to see from (3.12) that \( ||w^{k+1} - w^*||^2_{G_k} \) is bounded, where \( G_k \) is defined as (3.17). It shows that
\[
||x^{k+1} - x^*||^2_{T_k + \Sigma_f}, \quad ||y^{k+1} - y^*||^2_{S + \Sigma_g + \beta B^T B}, \quad ||\lambda^{k+1} - \lambda^*||^2
\]
are all bounded. Obviously, we claim that \( \{\lambda^k\} \) is bounded, and that \( \{y^k\} \) is bounded as \( S + \Sigma_g + \beta B^T B \) is assumed to be positive definite. Note that \( Ax^* + By^* - b = 0 \) and
\[
||Ax^{k+1} - Ax^*|| = ||Ax^{k+1} + By^{k+1} - b - B(y^{k+1} - y^*)|| \leq ||Ax^{k+1} + By^{k+1} - b|| + ||B(y^{k+1} - y^*)||.
\]
It then follows from (3.18) that \( ||A(x^{k+1} - x^*)|| \) is bounded, so is \( ||x^{k+1} - x^*||^2_{T_k + \Sigma_f + \beta A^T A} \). This shows that \( \{x^k\} \) is also bounded since \( T + \Sigma_f + \beta A^T A \) is assumed to be positive definite and \( T_k + \Sigma_f + \beta A^T A \succeq T + \Sigma_f + \beta A^T A \). Therefore, the sequence \( \{w^k\} \) is bounded.

Next we show that any cluster point of the sequence \( \{w^k\} \) is an optimal solution of (1.1) and the sequence \( \{w^k\} \) has only one cluster point. This can be done in a way similar to the proof of [9, Theorem 2.7].

3.3 Global Convergence of Algorithms 1 and 2

We establish the global convergence of Algorithms 1 and 2 as a corollary of the convergence results in subsection 3.2.

Throughout this subsection, let \( H^i_k = (B^i_k)^{-1} \) \((i = 1, 2)\). For the global convergence we need
\[
T^1_k = B^1_k - M^1 \succeq 0, \quad \text{that is} \quad B^1_k \succeq M^1,
\]
and
\[
T^2_k = B^2_k - M^2 \succeq 0, \quad \text{that is} \quad B^2_k \succeq M^2,
\]
where \( M^1 = \nabla^2_{xx} L_{\beta}(x, y, \lambda) = \beta A^T_1 A_1 \), and \( M^2 = \nabla^2_{xx} L_{\beta}(x, y, \lambda, \mu) = \beta_1 A^T_1 A_2 + \beta_2 I \).

From Subsection 2.2, we know that \( M^{\delta_1} = M^1 + \delta_1 I \succeq 0 \) with \( \delta_1 > 0 \), and \( B^1 \succeq M^{\delta_1} \succeq M^1 \). We need to show \( B^1_k \succeq M^{\delta_1} \), then \( T^1_k \succeq \delta_1 I \) for all \( k \). Note that \( B^1_k \succeq M^{\delta_1} \) is equivalent to \( H^1_k \succeq (M^{\delta_1})^{-1} \). We will show that \( H^1_k \succeq (M^{\delta_1})^{-1} \) for all \( k \) when the initial matrix \( H^0_0 \) satisfies
\[
H^0_0 \succeq (M^{\delta_1})^{-1}.
\]

Similarly from Subsection 2.3, it is necessary to show for all \( k \), \( B^2_k \succeq M^2 \), then \( T^2_k \succeq 0 \). The same as the above, \( B^2_k \succeq M^2 \) is equivalent to \( H^2_k \succeq (M^2)^{-1} \). We will show that \( H^2_k \succeq (M^2)^{-1} \) for all \( k \) when the initial matrix \( H^0_0 \) satisfies
\[
H^0_0 \succeq (M^2)^{-1}.
\]

Theorem 2.2 shows that if \( H_0 \succeq M^1 \), then \( H_k \succeq M^1 \), and hence \( T^1_k \succeq 0 \). Since that \( M^{\delta_1} \) and \( M^2 \) are constant positive definite matrixes, Theorem 2.2 is obviously suitable for \( M^{\delta_1} \) and \( M^2 \). Thus \( T^1_k \succeq \delta_1 I \) and \( T^2_k \succeq 0 \), respectively.

Now we give the convergence of Algorithm 1 for problem (1.7) (ADM-B1) and Algorithm 2 for problem (1.8) (ADM-B2).
Theorem 3.7. Suppose that the sequences $\{T_k^1\}$ and $\{\gamma_k^1\}$ satisfy the Condition 3.1. Let $w^*_1$ be the KKT points of problem (1.9), and let $\{w^k_1\}$ be generated by the Algorithm 1. Then the sequence $\{w^k_1\}$ converges to a solution of (1.9).

Theorem 3.8. Suppose that the sequences $\{T_k^2\}$ and $\{\gamma_k^2\}$ satisfy the Condition 3.1. Let $w^*_2$ be the KKT points of problem (1.10), and let $\{w^k_2\}$ be generated by the Algorithm 2. Then the sequence $\{w^k_2\}$ converges to a solution of (1.10).

Proof. The convergence in Theorems 3.7 and 3.8 directly follows from Theorem 3.6.

Remark 3.9. If there exists a constant $\bar{k}^1, 0 < \bar{k}^1 < \infty$, updating of $B^1_k$ by the BFGS procedure stops at $\bar{k}^1$. Then the sequences $\{T_k^1\}$ and $\{\gamma_k^1\}$ in Theorem (3.7) directly satisfy Condition 3.1. If there exists a finite constant $\bar{k}^2$, and updating of $B^2_k$ by the BFGS procedure stops at $\bar{k}^2$, the sequences $\{T_k^2\}$ and $\{\gamma_k^2\}$ in Theorem (3.8) easily satisfy Condition 3.1 as well.

4 Numerical Experiments for L1 regularized logistic regression model

In this section, we test the proposed ADMM by solving a popular sparse learning problem, L1 regularized logistic regression model:

$$
\min \left\{ \frac{1}{m} \sum_{i=1}^{m} \log \left( 1 + \exp \left( -r_i (A_i x + x_0) \right) \right) + \rho \|x\|_1 \mid x \in \mathbb{R}^n \right\},
$$

(4.1)

where $A \in \mathbb{R}^{m \times n}$ is a feature matrix, $A_i \in \mathbb{R}^{1 \times n}$ is the row vector of matrix $A$ and $r \in \mathbb{R}^m$ is a response vector. The scalar $m$ is the number of data points and $n$ is the dimension of data. $x_0 \in \mathbb{R}$ is a decided intercept scalar, and $\rho > 0$ is a regularization parameter. The decision variable of (4.1) is $x \in \mathbb{R}^n$. Note that the vector $x$ is the coefficient vector.

As mentioned above, our purpose is to justify the advantages of the ADMM with BFGS update (Algorithm 1). We choose some classical ADMMs as the benchmark for numerical comparison in the subsection.

By introducing some auxiliary variables $y_i \in \mathbb{R}$ ($i = 1, 2, ..., m$) and $z \in \mathbb{R}^n$, the L1 regularized logistic regression model (4.1) can be reformulated as

$$
\begin{align*}
\text{minimize} & \quad g(\tilde{y} = \left( \begin{array}{c} y \\ z \end{array} \right), \tilde{A} = \left[ \begin{array}{c} A \\ I_n \end{array} \right], B = -I_{m+n}, b = 0, \\
\text{subject to} & \quad y_i = A_i x, \quad i = 1, 2, ..., m \\
& \quad z = x, \\
& \quad x, z \in \mathbb{R}^n, y_i \in \mathbb{R}.
\end{align*}
$$

(4.2)

Note that, letting $\tilde{y} = \left( \begin{array}{c} y \\ z \end{array} \right)$, and $\tilde{A} = \left[ \begin{array}{c} A \\ I_n \end{array} \right]$, $B = -I_{m+n}, b = 0$, this problem is reduced to (1.1).

The augmented Lagrangian function of (4.2) can be written as:

$$
L_\beta(x, y, z, \lambda, \mu) = g(\tilde{y}) - \langle \lambda, A x - y \rangle - \langle \mu, x - z \rangle + \frac{\beta}{2} \|A x - y\|^2 + \frac{\beta}{2} \|x - z\|^2,
$$

(4.3)

where $\lambda \in \mathbb{R}^m, \mu \in \mathbb{R}^n$ are multipliers associated to the linear constraints and $\beta > 0$ is the penalty parameters. We further define $\tilde{\lambda} = \left( \begin{array}{c} \lambda \\ \mu \end{array} \right)$. Let $M$ be defined as the Hessian matrix of the augmented Lagrangian function (4.3), that is, $M = \nabla^2_{xx} L_\beta(x, y, z, \lambda, \mu) = \beta \tilde{A}^T \tilde{A} = \beta A^T A + \beta I$. Note that $M > 0$ whenever $\beta > 0$. Therefore, we set $\delta_1 = 0$ in Algorithm 1.

We test the following four methods:
\textbf{ADMM-1}: the classical ADMM [6, 7] which is applied for the original problem (4.1):
\[
\begin{align*}
    x^{k+1} &= \arg \min_x \frac{1}{m} \sum_{i=1}^m \log \left( 1 + \exp \left( - r_i (A_i x + x_0) \right) \right) + \frac{\beta}{2} \| x - y^k - \lambda^k / \beta \|_2^2, \\
    y^{k+1} &= S_{\rho / \beta} (x^{k+1} - \lambda^k / \beta), \\
    \lambda^{k+1} &= \lambda^k - \beta (x^{k+1} - y^{k+1}).
\end{align*}
\tag{4.4}
\]

\textbf{ADMM-2}: the classical ADMM [6, 7] applied for the problem (4.2);

\textbf{ADMM-PRO}: normal proximal ADMM [5] applied for the problem (4.2);

\textbf{ADM-BFGS (or ADM-LBFGS)}: the proximal ADMM with BFGS (or LBFGS) update for the problem (4.2):
\[
\begin{align*}
    x^{k+1} &= x^k + (B_k)^{-1} \left( A^T \lambda^k + \mu^k + \beta A^T y^k + \beta z^k - M x^k \right), \\
    y^{k+1} &= \arg \min_y \frac{1}{m} \sum_{i=1}^m \log \left( 1 + \exp \left( - r_i (y_i + x_0) \right) \right) + \frac{\beta}{2} \| A x^{k+1} - y - \lambda^k / \beta \|_2^2, \\
    z^{k+1} &= S_{\rho / \beta} (x^{k+1} - \mu^k / \beta), \\
    \lambda^{k+1} &= \lambda^k - \beta (A x^{k+1} - y^{k+1}), \\
    \mu^{k+1} &= \mu^k - \beta (x^{k+1} - z^{k+1}).
\end{align*}
\tag{4.5}
\]

Note that the above operator \( S_\kappa : \mathbb{R}^n \rightarrow \mathbb{R}^n \) is the soft-thresholding operator defined as
\[
(S_\kappa (a))_i = (1 - \kappa / |a_i|)_{+} \cdot a_i, \quad i = 1, ..., m,
\]
for \( \kappa > 0 \) and \( a \in \mathbb{R}^m \).

Since the \( x \)-subproblems in (4.4) and \( y \)-subproblems in (4.5) have no closed-form solution, we adopt a custom Newton solver for these subproblems with the tolerance of \( 10^{-6} \). We set a maximum iteration number of the Newton solver to 50. The codes for the original ADMM (4.4) and Newton solver are referred to the paper [2] and can be found at [1].

Now we specify the setting for the L1 regularized logistic regression model (4.1) to be tested. For the model, we generate data by the codes of [1]. The details are following as

\begin{itemize}
    \item the intercept \( x_0 \) is chosen from \( \mathcal{N}(0, 1) \);
    \item \( D \in \mathbb{R}^{m \times n} \) is a sparse matrix normally distributed with \( p \) sparsity nonzero entries, where \( p \in (0, 1) \);
    \item the vector \( r \) is generated by \( r = \text{sign}(Dw + x_0 + \epsilon) \), where \( \epsilon \) is the noise drawn from \( \mathcal{N}(0, 0.1) \), and \( w \in \mathbb{R}^n \) is a random and sparse vector with approximately 10% normally distributed nonzero entries;
    \item matrix \( A \) can be written as \( A = \text{spdiags}(r, 0, m, m) \cdot D \) by MATLAB, which is the product of a banded sparse matrix with \( D \);
    \item \( \rho = 0.1 \rho_{\max} \), where \( \rho_{\max} \) is the maximum regularization parameter when the solution is \( x^* = 0 \). The concrete definition of \( \rho_{\max} \) can be found in [12, Subsection 2.1];
    \item the initial points are \( x^0 = y^0 = z^0 = 0 \) and \( \lambda^0 = \mu^0 = 0 \).
\end{itemize}

We adopt the stopping criterion as in [2] that:
\[
\| \tilde{A} x^k + B y^k \|_2 \leq \epsilon_k^\text{pri} \quad \text{and} \quad \| \beta \tilde{A}^T B (\tilde{y}^k - y^{k-1}) \|_2 \leq \epsilon_k^\text{dual},
\tag{4.6}
\]
i.e., the primal residual and dual residual at the iteration \( k \) is small; \( \epsilon_k^\text{pri} > 0 \) and \( \epsilon_k^\text{dual} > 0 \) are feasibility tolerances for the primal and dual feasibility conditions, respectively. These tolerances can be chosen using an absolute and relative criterion from the suggestion in [2], such as
\[
\epsilon_k^\text{pri} = \sqrt{m} \epsilon_{\text{abs}}^\text{rel} + \epsilon_{\text{rel}}^\text{max} \{ \| \tilde{A} x^k \|_2, \| B y^k \|_2 \},
\]
\[
\epsilon_k^\text{dual} = \sqrt{m} \epsilon_{\text{abs}}^\text{rel} + \epsilon_{\text{rel}} \| \tilde{A}^T \lambda_k \|_2,
\]
where $\epsilon_{\text{abs}} > 0$ is an absolute tolerance and $\epsilon_{\text{rel}} > 0$ is a relative tolerance. The choice of stopping criterion depends on the scale of the variable values. Here are chosen as $\epsilon_{\text{abs}} = 10^{-4}$, $\epsilon_{\text{rel}} = 10^{-3}$.

The proximal terms chosen as follows:

ADM-PRO: An semidefinite proximal matrix $T$ as

$$T = \xi I - \beta I - \beta A^T A,$$

with $\xi = 1.01 \cdot \lambda_{\text{max}} (\beta I + \beta A^T A)$.

ADM-BFGS and ADM-LBFGS: An semidefinite proximal matrix sequence $T_k$ with $P_{k-1}$ generated by BFGS (2.1), the initial matrix

$$B_0 = \gamma I, \gamma = 1.01 \cdot \lambda_{\text{max}} (\beta I + \beta A^T A).$$

The penalty parameter $\beta$ is 1.

First, we set the sparsity of the matrix $D$ as $p = 0.1$. Table 1 shows the iterative steps and the CPU time (in seconds) of classical ADMM, proximal ADMM and ADMM with BFGS. The “Int.Iter.” in the table means the total internal iterations of the Newton method.

Table 1: Comparison on iteration steps and CPU time (seconds) among the four methods

<table>
<thead>
<tr>
<th>Dim</th>
<th></th>
<th>ADMM-1</th>
<th></th>
<th>ADMM-2</th>
<th></th>
<th>ADM-PRO</th>
<th></th>
<th>ADM-BFGS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>200</td>
<td>52.0</td>
<td>260.0</td>
<td>1.24</td>
<td>106.0</td>
<td>414.0</td>
<td>0.19</td>
<td>276.0</td>
<td>1074.0</td>
</tr>
<tr>
<td>500</td>
<td>500</td>
<td>65.0</td>
<td>390.0</td>
<td>6.80</td>
<td>170.0</td>
<td>675.0</td>
<td>0.64</td>
<td>588.0</td>
<td>2342.0</td>
</tr>
<tr>
<td>1000</td>
<td>500</td>
<td>115.0</td>
<td>690.0</td>
<td>32.57</td>
<td>143.0</td>
<td>569.0</td>
<td>0.76</td>
<td>394.0</td>
<td>1569.0</td>
</tr>
</tbody>
</table>

It is obvious to see from the above table that the classical ADMM-2 admits the faster method to catch the solution, while the ADMM with BFGS is better than the proximal ADMM on the number of iterations. We also know that classical ADMM-1 takes a lot of time computing the $x$-subproblems by the Newton method and that in classical ADMM-2 is easier, instead by computing the inverse of Hessian matrix. When the data matrix $A$ is ill condition or the inverse of Hessian matrix of augmented Lagrangian function is impossible to be computed, it is meaningful to use the proximal matrix $H_k$. ADMM with BFGS update needs more memory to save the $H_k$, thus we next will consider to use the L-BFGS to construct the $H_k$ for some larger cases. The memory size can be chosen as some certain $m$.

Here we take the memory as 20 for ADMM with L-BFGS update. We set the maximum outer iteration step is 2000 for all the methods. The results provided in Table 2 show the iteration steps and CPU time (in seconds) for different size and sparsity matrix $D$. We plot the objective function values with respect to the CPU time of four different size matrixes $D$ (same as matrix $A$) with sparsity $p = 10/n$ in Figure 1. In Figure 2 we give the histogram of CPU time with respect to the matrix size of $D$ with sparsity $p = 0.1$ in order to see the comparison of computing time clearly.

Table 2: Comparison on iteration steps and CPU time (seconds) among the four methods

<table>
<thead>
<tr>
<th>Dim</th>
<th></th>
<th>ADMM-1</th>
<th></th>
<th>ADMM-2</th>
<th></th>
<th>ADM-PRO</th>
<th></th>
<th>ADM-LBFGS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>500</td>
<td>23.0</td>
<td>115.0</td>
<td>2.26</td>
<td>233.0</td>
<td>927.0</td>
<td>1.24</td>
<td>488.0</td>
<td>1940.0</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>20.0</td>
<td>100.0</td>
<td>4.98</td>
<td>128.0</td>
<td>502.0</td>
<td>1.85</td>
<td>350.0</td>
<td>1370.0</td>
</tr>
<tr>
<td>1000</td>
<td>1500</td>
<td>22.0</td>
<td>109.0</td>
<td>11.39</td>
<td>132.0</td>
<td>515.0</td>
<td>4.43</td>
<td>352.0</td>
<td>1369.0</td>
</tr>
<tr>
<td>1000</td>
<td>2000</td>
<td>36.0</td>
<td>180.0</td>
<td>32.11</td>
<td>178.0</td>
<td>701.0</td>
<td>11.05</td>
<td>447.0</td>
<td>1753.0</td>
</tr>
<tr>
<td>1000</td>
<td>500</td>
<td>0.10</td>
<td>115.0</td>
<td>33.66</td>
<td>143.0</td>
<td>569.0</td>
<td>0.76</td>
<td>394.0</td>
<td>1569.0</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>0.10</td>
<td>143.0</td>
<td>859.0</td>
<td>114.26</td>
<td>161.0</td>
<td>639.0</td>
<td>2.41</td>
<td>882.0</td>
</tr>
<tr>
<td>1000</td>
<td>1500</td>
<td>0.10</td>
<td>171.0</td>
<td>1027.0</td>
<td>273.72</td>
<td>188.0</td>
<td>745.0</td>
<td>6.78</td>
<td>1361.0</td>
</tr>
<tr>
<td>1000</td>
<td>2000</td>
<td>0.10</td>
<td>160.0</td>
<td>961.0</td>
<td>424.03</td>
<td>238.0</td>
<td>949.0</td>
<td>15.94</td>
<td>1701.0</td>
</tr>
</tbody>
</table>
Figure 1: $p = 10/n$: Evolution of the objective function values with respect to CPU time
From Table 2 and Figures 1 and 2, we conclude that ADM-LBFGS performs better and stability. When the matrix $A$ is larger and has more non-zero elements, ADMM-1 spends too much time and ADMM-2 spends less time than ADMM-1 but more than ADM-LBFGS. The usual proximal ADMM (ADM-PRO) takes much more iteration steps. ADM-LBFGS algorithm can reach the optimal solution fast both at iteration and CPU time. Thus, if the data matrix $A$ is large enough or ill condition (it is difficult or impossible to compute the inverse of matrix $\beta I + \beta A^\top A$), it is better to choose ADM-LBFGS method.

5 Conclusions

In this paper, we considered a special proximal ADMM where the proximal matrix derived from the BFGS or Limited memory BFGS method for the general convex optimization problems (1.7) and (1.8). The $x$-subproblems of these convex problems can be rewritten as unconstrained quadratic programming problems in Subsection 2.2 and 2.3 as that in paper [9], and hence the Hessian matrix of the augmented Lagrangian function is a constant matrix. The global convergence of such methods have also been established under some standard conditions. The numerical results for L1 regularized logistic regression problem are given to show the feasibility and effectiveness of the proposed algorithms.

References


