Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 4864296, 7 pages
https://doi.org/10.1155/2020/4864296

Research Article

Alternating Direction Multiplier Method for Matrix $l_{2,1}$-Norm Optimization in Multitask Feature Learning Problems

Yaping Hu, Liying Liu, and Yujie Wang

College of Science, Tianjin University of Science and Technology, Tianjin, China

Correspondence should be addressed to Yaping Hu; huyaping@tust.edu.cn

Received 6 July 2020; Accepted 6 August 2020; Published 26 August 2020

Academic Editor: Gonglin Yuan

The joint feature selection problem can be resolved by solving a matrix $l_{2,1}$-norm minimization problem. For $l_{2,1}$-norm regularization, one of the most fascinating features is that some similar sparsity structures can be employed by multiple predictors. However, the nonsmooth nature of the problem brings great challenges to the problem. In this paper, an alternating direction multiplier method combined with the spectral gradient method is proposed for solving the matrix $l_{2,1}$-norm optimization problem involved with multitask feature learning. Numerical experiments show the effectiveness of the proposed algorithm.

1. Introduction

Because of its widespread application in high-dimensional sparse learning, the feature selection problem has been concerned widely by the machine learning community in multitask feature learning and become a hot research field in recent years. The purpose of multitask feature learning is to learn the shared information between related tasks, so as to promote the learning effect. Learning multiple related tasks simultaneously is much more efficient compared to single-task learning particularly [1, 2]. For feature selection tasks in multitask learning, using mixed $l_{2,1}$-norm can produce joint sparsity in the feature layer and task layer. In particular, $l_{2,1}$-norm is sometimes more advantageous because it often leads to more sparse solutions. In multitask learning by Obosinski et al. [3] and Argyriou et al. [4], the regularization of $l_{2,1}$-norm is introduced for the first time. In recent years, a lot of research studies have been carried out on it. Multiple predictors are encouraged to share similar parameter sparse patterns from different tasks [3–5], which is a very attractive feature of the $l_{2,1}$-norm regularized problem. When the objective function is convex, the $l_{2,1}$-norm regularized problem is convex and has a global optimal solution. However, the optimization problem is difficult to solve because of the nonsmoothness of $l_{2,1}$-norm regularization. The method in Liu et al. [6] transforms the minimization problem of $l_{2,1}$-norm into two equivalent convex smooth optimization problems and then minimizes them by Nesterov's accelerated gradient method [7]. For the $l_{2,1}$-norm regularized problem, a proximal alternating direction method was presented recently by Xiao et al. [8]. Hu et al. [9] proposed inexact accelerated proximal gradient algorithms to solve the $l_{2,1}$-norm regularization.

The training set of k tasks is given by $\{(a_i^j, b_i^j)\}_{i=1}^{m_j}$ $(j = 1, 2, \ldots, t)$, where for the j^{th} task, $a_i^j \in \Re^n$ denotes the i^{th} sample, m_j denotes the number of training samples, b_i^j denotes the corresponding response, and the total number of training samples is $m = \sum_{j=1}^{t} m_j$. The matrix $A_j = [a_1^j, \ldots, a_{m_j}^j]^T \in \Re^{m_j \times n}$ is the data for the j^{th} task, and $b_j = [b_1^j, \ldots, b_{m_j}^j]^T \in \Re^{m_j}$, $X_{:,j} \in \Re^n$ is the sparse feature for the j^{th} task. $A = [A_1, \ldots, A_t] \in \Re^{m \times n}$, $b = [b_1, \ldots, b_t]^T \in \Re^m$, and $X = [X_{:,1}, \ldots, X_{:,t}] \in \Re^{n \times t}$ are the joint learning features for the task in multitask learning. It is encouraged to set elements of several rows in X to be zero to select features globally. According to Argyriou et al. [4], the problem of $l_{2,1}$-norm minimization can be described as

$$\min_{X \in \Re^{n \times t}} \frac{1}{2} \sum_{j=1}^{t} \left\| A_j X_{:,j} - b_j \right\|_2^2 + \mu \|X\|_{2,1}, \tag{1}$$

in which matrix $\|X\|_{2,1}$ is defined as

$$\|X\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{t} X_{i,j}^2} = \sum_{i=1}^{n} \left\|X_{i,:}\right\|_2, \tag{2}$$

where $X_{i,:}$ denotes the i^{th} row element of matrix X and $X_{:,j}$ denotes the j^{th} column element of matrix X. The first term measures the loss caused by matrix X which is based on the training data samples of A and b, and the second term is the regularization term in (1), where $\mu > 0$ is the regularization parameter which can be used to keep a balance between the two terms to minimize.

As described in [10], the alternate direction multiplier method (ADMM) is a natural method in the field of large-scale data distribution machine learning and big data-related optimization because it can process the objective function separately and synchronously, and it has aroused widespread attention in the past few years. ADMM method is widely used in a lot of fields, such as image restoration [11], machine learning [12], and compressed sensing [13]. This widespread application has sparked a strong interest in further understanding the theoretical nature of the ADMM (see [14–17]).

Barzilai and Borwein in [18] first proposed the spectral gradient method to solve the strict convex quadratic minimization problems. Due to efficiency and computational cheapness, BB method has caused wide attention in the area of optimization. Raydan [19] developed this method to solve general unconstrained optimization problems. Recently, the BB method has been successfully extended for solving the nonsmooth convex optimization problem [20].

In this paper, an ADMM with the spectral gradient method is proposed to solve the $l_{2,1}$-norm regularization problem in the area of multitask learning. We first add a new auxiliary variable to the augmented Lagrangian form of (1), then iteratively minimize the augmented Lagrangian function in which an exact method is used to solve one subproblem, and the spectral gradient method is employed to solve the other subproblems. Experimental results show that the proposed ADMM-BB method is competitive, fast, and efficient.

The rest of the paper is arranged as follows. Section 2 introduces the ADMM method for solving (1). Section 3 explains how to find the solution to the subproblems generated by each iteration and gives a practical ADMM using the spectral gradient algorithm. Section 4 gives the numerical results of the simulation data set and the real data set and compares them with other methods. Finally, Section 5 summarizes and concludes this article.

2. ADMM for $l_{2,1}$-Norm Minimization

The $l_{2,1}$-norm matrix minimization problem has the following standard form:

$$\min_{X \in \Re^{n \times t}} \frac{1}{2}\|\mathscr{A}(X) - b\|_2^2 + \mu\|X\|_{2,1}, \tag{3}$$

where $\mathscr{A}: \Re^{n \times t} \longrightarrow \Re^m$ is a mapping defined based on matrix vector multiplication for each learning task, i.e., $\mathscr{A}(X) = [A_1 X_{:,1}, \ldots, A_t X_{:,t}] \in \Re^m$. By introducing auxiliary variable Y, problem (3) is equivalently transformed into a linearly constrained convex programming problem:

$$\begin{aligned} \min_{X \in \Re^{n \times t}} \quad & \frac{1}{2}\|\mathscr{A}(Y) - b\|_2^2 + \mu\|X\|_{2,1} \\ \text{s.t.} \quad & X - Y = 0. \end{aligned} \tag{4}$$

The augmented Lagrangian function of problem (4) is defined as

$$\begin{aligned} \mathscr{L}_c(X, Y, Z) = & \frac{1}{2}\|\mathscr{A}(Y) - b\|_2^2 + \mu\|X\|_{2,1} \\ & - \langle Z,\ X - Y\rangle + \frac{c}{2}\|X - Y\|^2, \end{aligned} \tag{5}$$

where $c > 0$ is the penalty parameter, $\langle \cdot \rangle$ means the standard trace inner product, $\langle X, Y\rangle := \mathrm{Tr}(X^T Y)$ for X and Y in $\Re^{n \times t}$, symbol "Tr" represents the trace, i.e., the sum of the diagonal elements of a squared matrix which is also equal to the sum of the eigenvalues. For any matrix $X \in \Re^{n \times t}$, $\|\cdot\|$ is defined as the Frobenius norm:

$$\|X\| = \|X\|_{Fr} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{t} x_{i,j}^2}, \tag{6}$$

where $x_{i,j}$ is the $(i, j)^{\text{th}}$ element of matrix X so that $\|X\| = \sqrt{\langle X, X\rangle}$. For solving (5), the iterative scheme of the alternating direction method of multipliers is

$$X^{k+1} = \arg\min_{X \in \Re^{n \times t}} \left\{\mu\|X\|_{2,1} - \langle Z^k, X\rangle + \frac{c}{2}\left\|X - Y^k\right\|^2\right\}, \tag{7}$$

$$Y^{k+1} = \arg\min_{Y \in \Re^{n \times t}} \left\{\|\mathscr{A}(Y) - b\|_2^2 + \langle Z^k, Y\rangle + \frac{c}{2}\left\|X^{k+1} - Y\right\|^2\right\}, \tag{8}$$

$$Z^{k+1} = Z^k - c\left(X^{k+1} - Y^{k+1}\right), \tag{9}$$

where $Z^k \in \Re^{n \times t}$ is the Lagrange multiplier.

The alternating direction multiplier method for solving problem (4) can be expressed as follows.

Algorithm 1. ADMM for $l_{2,1}$-norm minimization problem.

Step 1: find X^{k+1} via

$$0 \in \partial\left(\mu\left\|X^{k+1}\right\|_{2,1}\right) - \left[Z^k - c\left(X^{k+1} - Y^k\right)\right], \tag{10}$$

where $\partial(\cdot)$ represents the subgradient operator of the convex function $\|\cdot\|_{2,1}$.

Step 2: solve Y^{k+1} via

$$\langle Y - Y^{k+1}, \mathscr{A}^*\left(\mathscr{A}\left(Y^{k+1}\right) - b\right) + Z^k - c\left(X^{k+1} - Y^{k+1}\right)\rangle \geq 0, \quad \forall Y \in \mathfrak{R}^{n\times t}. \tag{11}$$

Step 3: compute the multiplier Z^{k+1} by (9).

The following result shows that the optimal solution set of $l_{2,1}$-norm matrix minimization problem (3) is bounded (see [9]).

Lemma 1. *For each $\mu > 0$, the optimal solution set $\mathscr{X}^*$ of (3) is bounded, and for any $X^* \in \mathscr{X}$, we have*

$$\|X\| \leq \chi. \tag{12}$$

$$\chi = \begin{cases} \min\left\{\dfrac{\|b\|_2^2}{(2\mu)}, \left\|\mathscr{A}^*\left(\mathscr{A}\mathscr{A}^*\right)^{-1} b\right\|_{2,1}\right\}, & \mathscr{A} \text{ is surjective,} \\ \dfrac{\|b\|_2^2}{(2\mu)}, & \text{otherwise.} \end{cases} \tag{13}$$

The global convergence property of Algorithm 1 holds directly based on the results developed by Bertsekas and Tsitsiklis [21], Chapter 3, p. 256 for general convex programming problems.

Theorem 1. *Let $\{(X^k, Y^k, Z^k)\}$ be the sequence generated by Algorithm 1 with $\lim_{k\longrightarrow\infty} c^k = c^\infty \leq \infty$. Then, $\{(X^k, Y^k, Z^k)\}$ is bounded, and every limit of $\{(X^k, Y^k)\}$ is an optimal solution of equivalent problem (4).*

3. ADMM-BB Method for $l_{2,1}$-Norm Minimization

Section 2 gives the theoretical alternating direction multiplier method of the $l_{2,1}$-norm minimization problem. However, a key issue has not yet been resolved: how to solve subproblems (7) and (8) efficiently? This problem is fundamentally important because if it is difficult to solve each subproblem, this method will not be useful anyway. In this paper, an exact method is used to solve (7), and the spectral gradient method is employed to solve (8).

Given $c > 0$ and $(Y^k, Z^k) \in (\mathfrak{R}^{n\times t}, \mathfrak{R}^{n\times t})$, we have

$$\begin{aligned} X^{k+1} &= \arg\min_{X\in\mathfrak{R}^{n\times t}} \mathscr{L}_c\left(X, Y^k, Z^k\right) \\ &= \arg\min_{X\in\mathfrak{R}^{n\times t}} \mu\|X\|_{2,1} - \langle Z^k, X - Y^k\rangle \\ &\quad + \frac{c}{2}\left\|X - Y^k\right\|^2 \\ &= \arg\min_{X\in\mathfrak{R}_{n\times t}} \mu\|X\|_{2,1} \\ &\quad + \frac{c}{2}\left\|X - \left(Y^k + \frac{1}{c}Z^k\right)\right\|^2. \end{aligned} \tag{14}$$

Let $N = Y^k + 1/cZ^k$. Equation (14) has the solution of the form

$$X^{k+1} = \arg\min_{X_{1,:},\ldots,X_{n,:}} \sum_{i=1}^{n}\left(\mu\left\|X_{i,:}\right\|_{2,1} + \frac{c}{2}\left\|X_{i,:} - N_{i,:}\right\|^2\right), \tag{15}$$

which indicates that involved problem (15) can be broken down into n independent t-dimensional subproblems:

$$\min_{X_{i,:}\in\mathfrak{R}^t} \mu\left\|X_{i,:}\right\|_{2,1} + \frac{c}{2}\left\|X_{i,:} - N_{i,:}\right\|^2, \quad i = 1, 2, \ldots, n. \tag{16}$$

Clearly, the optimal solution $X_{i,:}^*$ can be obtained in the direction $N_{i,:}$, and it has the form of the formula $X_{i,:}^* = aN_{i,:}$ in which $a \geq 0$ is a parameter. Based on developing a Lagrangian dual form, subproblem (16) has a closed-form solution (see, e.g., [22, 23]) which can be explicitly expressed as

$$X_{i,:}^* = \left(1 - \frac{\mu}{c\left\|N_{i,:}\right\|_2}\right)_+ N_{i,:}, \quad i = 1, \ldots, n, \tag{17}$$

where $(\cdot)_+ = \max(\cdot, 0)$. Therefore, the closed-form solution of (10) is given as follows:

$$X^{k+1} = \begin{bmatrix} \left(1 - \dfrac{\mu}{c\left\|\left(Y^k + 1/cZ^k\right)_{1,:}\right\|_2}\right) + \left(Y^k + \dfrac{1}{c}Z^k\right)_{1,:} \\ \left(1 - \dfrac{\mu}{c\left\|\left(Y^k + 1/cZ^k\right)_{2,:}\right\|_2}\right) + \left(Y^k + \dfrac{1}{c}Z^k\right)_{2,:} \\ \cdots \\ \left(1 - \dfrac{\mu}{c\left\|\left(Y^k + 1/cZ^k\right)_{n,:}\right\|_2}\right) + \left(Y^k + \dfrac{1}{c}Z^k\right)_{n,:} \end{bmatrix}. \tag{18}$$

Next, we analyze another subproblem (8). For fixed $c > 0$, let

$$\theta(Y) := \mathscr{L}_c\left(X^{k+1}, Y, Z^k\right). \tag{19}$$

Now, we investigate how to use the spectral gradient method to solve the corresponding problem:

$$\min_{Y\in\mathfrak{R}^{n\times t}} \theta(Y). \tag{20}$$

The function $\theta(Y)$ is convex and everywhere differentiable with

$$\nabla\theta(Y) = \mathscr{A}^*\left(\mathscr{A}(Y) - b\right) + Z^k - c\left(X^{k+1} - Y\right). \tag{21}$$

In order to distinguish the superscript in Algorithm 1, we apply the subscripts in the iteration of this subproblem. Spectral gradient method is defined by

$$Y_{j+1} = Y_j - \alpha_j \nabla\theta\left(Y_j\right), \tag{22}$$

where α_j is given by

$$\alpha_j = \frac{s_{j-1}^T u_{j-1}}{u_{j-1}^T u_{j-1}}, \tag{23}$$

where $s_{j-1} = Y_j - Y_{j-1}$ and $u_{j-1} = \nabla\theta(Y_j) - \nabla\theta(Y_{j-1})$.

Now, the spectral gradient method for (20) can be described as given in Algorithm 2.

Algorithm 2. The spectral gradient method.

Step 0: given $Y_0 = Y^k$, $\epsilon \in (0, 1)$, $\alpha_0 = 1$, and $j := 0$.

Step 1: termination criterion: stop if Y_j satisfies termination condition $\|\nabla\theta(Y_j)\| < \epsilon$. Otherwise, go to the next step.

Step 2: compute α_j by (23) if $j > 0$. Let $Y_{j+1} = Y_j - \alpha_j \nabla\theta(Y_j)$.

Step 3: let $j := j + 1$ and go to Step 1.

Finally, by adopting a relaxation factor γ, the multiplier update formula in Algorithm 1 is replaced by

$$Z^{k+1} = Z^k - \gamma c\left(X^{k+1} - Y^{k+1}\right), \quad 0 < \gamma < \frac{\sqrt{5}+1}{2}. \tag{24}$$

Glowinski in [24] first suggested the instruction of γ, and it has shown better performance in numerical experiments [25].

Now, a practical ready-to-implement version of the ADMM (7)–(9) can be described as follows.

Algorithm 3. ADMM-BB for the $l_{2,1}$-norm minimization problem.

Step 0: let c_0 and $\kappa > 1$ be given. Let Y^0 be arbitrary. Let Z^0 be the initial estimated Lagrange multipliers. Let $k := 0$.

Step 1: when the stopping criterion holds, then stop; otherwise, continue.

Step 2: compute X^{k+1} by (18).

Step 3: compute Y^{k+1} by solving the following problem with the spectral gradient method:

$$\min_{Y \in \Re^{m \times n}} \theta(Y). \tag{25}$$

Step 4: compute Z^{k+1} by (24).

Step 5: let $k := k + 1$ and go to Step 1.

Based on the conclusions of Bertsekas and Tsitsiklis ([21], Chapter 3, Proposition 4.2) and Glowinski ([24], Chapter VI, Theorem 5.1), for Algorithm 3, the following convergence conclusion holds.

Theorem 2. *Suppose that $\mathscr{L}_c$ has a saddle point $\{\overline{X}, \overline{Y}, \overline{Z}\}$. Let (X^k, Y^k, Z^k) be the sequence generated by Algorithm 3 with $c > 0$ and $\gamma \in (0, \sqrt{5} + 1/2)$. Then,*

$$\begin{aligned} &\lim_{k \longrightarrow \infty} \left(X^k, Y^k\right) = (\overline{X}, \overline{Y}), \\ &\lim_{k \longrightarrow \infty} \left(Z^{k+1} - Z^k\right) = 0, \\ &Z^k \text{ is bounded.} \end{aligned} \tag{26}$$

Moreover, if Z^* is a weak cluster point of $\{Z^k\}$, then $\{\overline{X}, \overline{Y}, Z^*\}$ is a saddle point of $\mathscr{L}_c$.

4. Experiments

In this section, we will give the numerical experimental results of ADMM-BB to solve matrix $l_{2,1}$-norm minimization problem (3). The experiments are carried out by MATLAB R2018b running on a computer with 2.8 GHz Intel Pentium CPU and 8 GB of low voltage memory.

Based on simulated data and real data, we conducted two types of numerical experiments to study the performance of the ADMM-BB method. In the test, we compared the ADMM-BB method with the IADM-MFL method [8] because the IADM-MFL method is well known and gives a feasible way to find a solution to the joint feature selection problem in the area of multitask learning. For each test function, starting with the origin point, when the distance between adjacent iteration points is less than a given constant tol, the algorithm stops, i.e.,

$$\text{RelChg} = \frac{\left\|X_k - X_{k-1}\right\|_F}{\left\|X_{k-1}\right\|_F} \leq \text{tol}. \tag{27}$$

We choose $c = 0.01/\text{mean}(|b|)$ in the following series of experiments.

Example 1. As [4], the simulation data sets are created by using a 5-dimensional zero-mean Gaussian distribution with a covariance matrix that equals to diag {1, 0.64, 0.49, 0.36, 0.25}, which can be denoted by $\overline{X}_{:,j}$. For all $\overline{X}_{:,j}$, expand it to 20 irrelevant dimensions by adding zero elements. The training data A_j are a random Gaussian matrix generated by Matlab command randn(m_j, n). Using A_j and $\overline{X}_{:,j}$, get the outputs b_j as

$$b_j = A_j \overline{X}_{:,j} + \omega, \tag{28}$$

where Gaussian noise ω is described by a mean of 0 and a standard deviation of $1.e-2$. In each performed method, X^* denotes the optimal solution of matrix $l_{2,1}$-norm minimization problem (3). To measure the quality of X^* to original $\overline{X}$, we set the relative error as follows:

$$\text{RelErr} = \frac{\left\|X^* - \overline{X}\right\|_F}{\left\|\overline{X}\right\|_F}. \tag{29}$$

We will analyze the performance of both methods with different number of dimensions and tasks because they will certainly affect the performance of each algorithm as an important factor. The numerical results are shown in Table 1, which contains the CPU time required in seconds (TIME), the total number of iterations (ITER), the total number of

Table 1: Numerical results for the random problem.

b, X	ADMM-BB			IADM-MFL		
(m, n, t)	ITER	TIME	RelErr	ITER	TIME	RelErr
(10000, 10, 100)	14	0.0506	$2.87e-03$	32	0.1719	$3.36e-03$
(20000, 10, 200)	15	0.2188	$2.65e-03$	31	0.2813	$2.97e-03$
(30000, 10, 300)	15	0.2344	$2.98e-03$	31	0.2969	$3.25e-03$
(10000, 15, 100)	18	0.0625	$3.75e-03$	42	0.2500	$3.49e-03$
(20000, 15, 200)	18	0.2969	$3.55e-03$	41	0.3906	$3.24e-03$
(30000, 15, 300)	19	0.3906	$3.66e-03$	44	0.4219	$3.47e-03$
(10000, 20, 100)	22	0.1406	$4.78e-03$	51	0.4375	$4.55e-03$
(20000, 20, 200)	22	0.3906	$4.92e-03$	52	0.5613	$4.27e-03$
(30000, 20, 300)	22	0.5938	$5.43e-03$	54	1.3906	$4.35e-03$
(10000, 25, 100)	27	0.2500	$6.49e-03$	66	0.5938	$4.94e-03$
(20000, 25, 200)	26	0.2969	$6.40e-03$	64	0.7656	$4.94e-03$
(30000, 25, 300)	26	0.7344	$6.81e-03$	65	1.5156	$5.11e-03$
(10000, 30, 100)	32	0.3594	$9.04e-03$	80	0.7188	$7.03e-03$
(20000, 30, 200)	31	0.6250	$9.23e-03$	81	0.8750	$6.27e-03$
(30000, 30, 300)	31	0.9688	$8.47e-03$	81	1.7344	$5.54e-03$

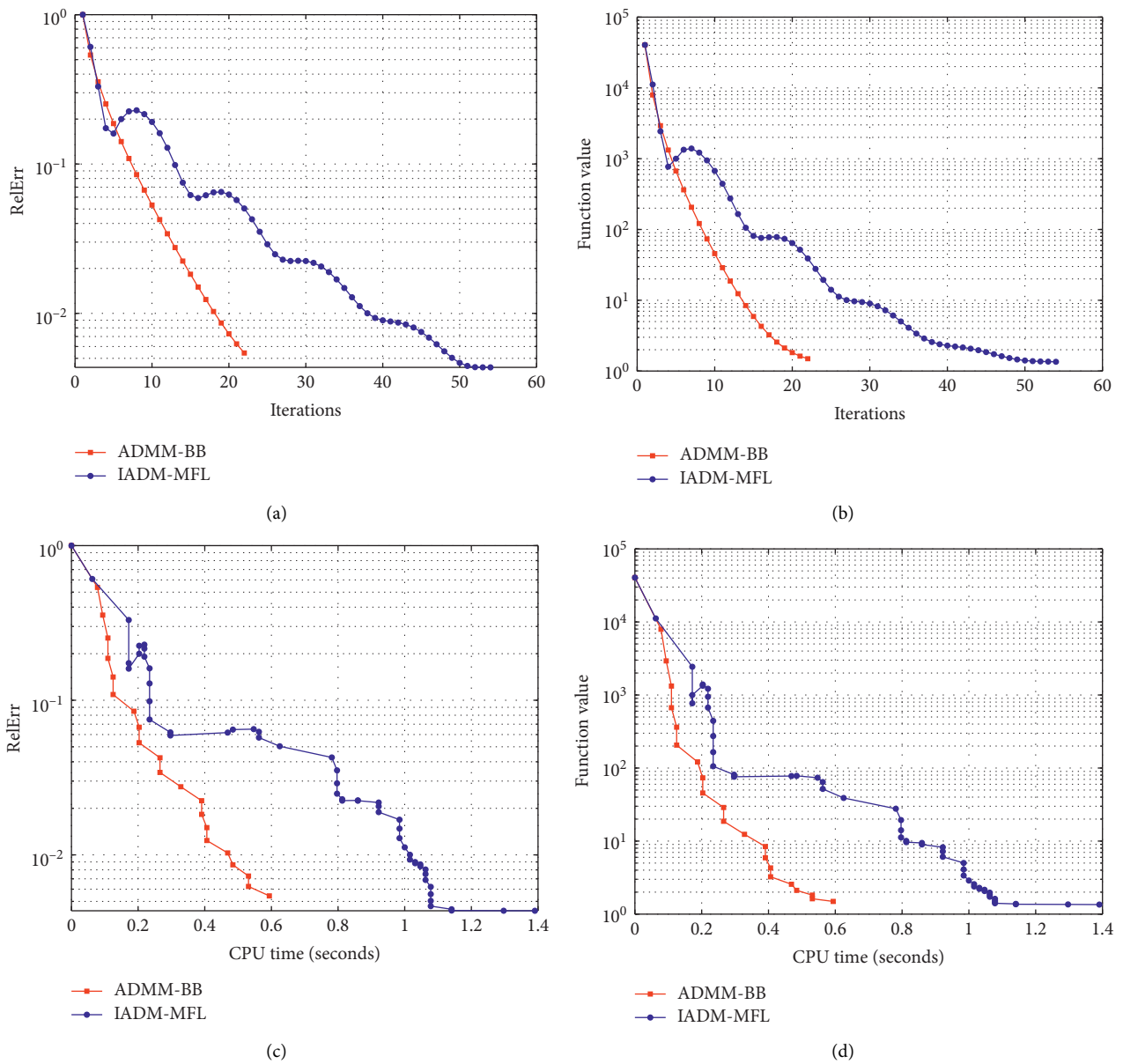

Figure 1: Convergence performance of ADMM-BB and IADM-MFL.

Table 2: Numerical results for the real problem.

Demo	b, X	ADMM-BB			IADM-MFL		
	(m, n, t)	ITER	TIME	RelChg	ITER	TIME	RelChg
20%	(80, 500, 10)	178	1.34	$9.95e-04$	365	2.47	$9.97e-04$
30%	(120, 500, 10)	282	2.23	$9.98e-04$	412	2.54	$9.93e-04$
40%	(160, 500, 10)	266	2.33	$9.96e-04$	415	2.73	$9.92e-04$
50%	(200, 500 10)	300	2.75	$9.95e-04$	404	2.80	$9.73e-04$
60%	(240, 500, 10)	306	2.86	$9.99e-04$	427	3.13	$9.95e-04$
70%	(280, 500, 10)	349	3.30	$9.99e-04$	460	3.47	$9.97e-04$

tasks (t), the dimension of the test data (n), and the dimension of the outputs (m).

From the results in Table 1, it can be seen that although both methods have successfully terminated, the number of iterations and CPU time of the ADMM-BB method are much less than IADM-MFL.

Then, the parameters involved in the methods are specified as $\text{tol} = 1e-3$, $t = 300$, $n = 20$, $\mu = 1e-2$, and $m_j = 100\,(j = 1, 2, \ldots, t)$. For each solving algorithm, we evaluate the objective function values and test error rate, and the convergence behavior of these algorithms is shown in Figure 1. To visually compare the convergence speed of the algorithms, the four subgraphs in Figure 1 show the change of the function value and relative error with the number of iterations and CPU time for both ADMM-BB and IADM-MFL algorithms. We present the relative error and the objective function values plotted against the number of iterations in the first row in Figure 1 and present the relative error and the objective function values plotted against the computational time in the second row. Figure 1 shows that although both ADMM-BB and IADM-MFL algorithms generate decreasing sequences and converge to the same function value as well as relative error, the performance of ADMM-BB is better than IADM-MFL in terms of iteration numbers and CPU time.

Example 2. In this test, we demonstrate the performance of the proposed algorithms on a real data set. dmoz is a text categorization data set, in which every 10 tasks correspond to subcategories of the arts category. The dmoz data set can be downloaded from http://www.dmoz.org/. In order to learn the joint feature between tasks, we randomly select data from each task for training and sample 20%, 30%, 40%, 50%, 60% and 70%, respectively, of the dmoz data set and then test the two methods at the same time. Except for $\mu = 1e-4$, the other parameters are the same as in the proceeding example for both ADMM-BB and IADM-MFL methods. The corresponding numerical results are summarized in Table 2.

From Table 2, we can see that ADMM-BB is an effective method and works better on these problems.

5. Conclusion

The convergence theory for the alternating direction multiplier method for the convex optimization problem has been well established by Bertsekas and Tsitsiklis [21] and Glowinski [24]. The main purpose of this paper is to demonstrate that this method is robust for the matrix $l_{2,1}$-norm regularized minimization problem. The key element is the practical efficiency of the alternating direction multiplier method by using the spectral gradient method in this paper. The corresponding numerical results verify the encouraging efficiency of the proposed method in solving the joint feature selection problem.

Data Availability

The data used to support the findings of this study are available in tables in this paper and can also be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Scientific Research Project of Tianjin Education Commission (no. 2019KJ232).

References

[1] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *Journal of Machine Learning Research*, vol. 6, pp. 1817–1853, 2005.

[2] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, pp. 615–637, 2005.

[3] G. Obozinski, B. Taskar, and M. I. Jordan, "Multi-Task Feature Selection," Technical Report, University of California, Berkeley, CA, USA, 2006.

[4] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[5] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint $l_{2,1}$-norms minimization," in *Proceedings of the Neural Information Processing Systems Foundation*, Vancouver, Canada, December 2010.

[6] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning Via efficient $l_{2,1}$-norm minimization," in *Proceedings of the UAI 2009 Conference*, Montreal, Canada, 2009.

[7] Y. Nesterov, *Gradient Methods for Minimizing Composite Objective Function*, Center for Operations Research and Econometrics (CORE), Louvain-la-Neuve, Belgium, 2007.

[8] Y. Xiao, S. Wu, S.-Y. Wu, and B.-S. He, "A proximal alternating direction method for $\ell_{2, 1} $-norm least squares problem in multi-task feature learning," *Journal of Industrial & Management Optimization*, vol. 8, no. 4, pp. 1057–1069, 2012.

[9] Y. Hu, Z. Wei, and G. Yuan, "Inexact accelerated proximal gradient algorithms for matrix $l_{2,1}$-norm minimization problem in multi-task feature learning," *Statistics, Optimization and Information Computing*, vol. 2, no. 4, pp. 352–367, 2014.

[10] 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*, vol. 3, no. 1, pp. 1–122, 2011.

[11] T. Goldstein and S. Osher, "The split bregman method for L1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[12] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 99, pp. 1663–1707, 2010.

[13] J. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1-Problems in compressive sensing," *SIAM Journal on Scientific Computing*, vol. 33, no. 1, pp. 250–278, 2011.

[14] G. Banjac, P. Goulart, B. Stellato, and S. Boyd, "Infeasibility detection in the alternating direction method of multipliers for convex optimization," *Journal of Optimization Theory and Applications*, vol. 183, no. 2, pp. 490–519, 2019.

[15] D. Boley, "Local linear convergence of the alternating direction method of multipliers on quadratic or linear programs," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2183–2207, 2013.

[16] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Journal of Scientific Computing*, vol. 66, no. 3, pp. 889–916, 2016.

[17] J. Jian, Y. Zhang, and M. Chao, "A regularized alternating direction method of multipliers for a class of nonconvex problems," *Journal of Inequalities and Applications*, vol. 193, 2019.

[18] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[19] M. Raydan, "The barzilai and borwein gradient method for the large scale unconstrained minimization problem," *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 26–33, 1997.

[20] G. Yuan and Z. Wei, "The barzilai and borwein gradient method with nonmonotone line search for nonsmooth convex optimization problems," *Mathematical Modelling and Analysis*, vol. 17, no. 2, pp. 203–216, 2012.

[21] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1989.

[22] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *Journal of Machine Learning Research*, vol. 10, pp. 2899–2934, 2009.

[23] M. Kowalski, "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303–324, 2009.

[24] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, pp. 168–179, Springer, New York City, NY, USA, 1984.

[25] B. He, S. L. Wang, and H. Yang, "A modified variable-penalty alternating directions method for monotone variational inequalities," *Journal of Computational Mathematics*, vol. 21, no. 4, pp. 495–504, 2003.