ACCELERATED LINEARIZED BREGMAN METHOD

BO HUANG[†], SHIQIAN MA[*†], AND DONALD GOLDFARB[†]

June 21, 2011

Abstract. In this paper, we propose and analyze an accelerated linearized Bregman (ALB) method for solving the basis pursuit and related sparse optimization problems. This accelerated algorithm is based on the fact that the linearized Bregman (LB) algorithm is equivalent to a gradient descent method applied to a certain dual formulation. We show that the LB method requires $O(1/\epsilon)$ iterations to obtain an ϵ-optimal solution and the ALB algorithm reduces this iteration complexity to $O(1/\sqrt{\epsilon})$ while requiring almost the same computational effort on each iteration. Numerical results on compressed sensing and matrix completion problems are presented that demonstrate that the ALB method can be significantly faster than the LB method.

Key words. Convex Optimization, Linearized Bregman Method, Accelerated Linearized Bregman Method, Compressed Sensing, Basis Pursuit, Matrix Completion

AMS subject classifications. 68U10, 65K10, 90C25

1. Introduction.

In this paper, we are interested in the following optimization problem

$$\min_{x\in\mathbb{R}^n} J(x) \quad \text{s.t.} \quad Ax = b, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $J(x)$ is a continuous convex function. An important instance of (1.1) is the so-called basis pursuit problem when $J(x) := \|x\|_1 = \sum_{j=1}^n |x_j|$:

$$\min_{x\in\mathbb{R}^n} \|x\|_1 \quad \text{s.t.} \quad Ax = b. \tag{1.2}$$

Since the development of the new paradigm of compressed sensing [9, 11], the basis pursuit problem (1.2) has become a topic of great interest. In compressed sensing, A is usually the product of a sensing matrix Φ and a transform basis matrix Ψ and b is a vector of the measurements of the signal $s = \Psi x$. The theory of compressed sensing guarantees that the sparsest solution (i.e., representation of the signal $s = \Psi x$ in terms of the basis Ψ) of $Ax = b$ can be obtained by solving (1.2) under certain conditions on the matrix Φ and the sparsity of x. This means that (1.2) gives the optimal solution of the following NP-hard problem [21]:

$$\min_{x\in\mathbb{R}^n} \|x\|_0 \quad \text{s.t.} \quad Ax = b, \tag{1.3}$$

where $\|x\|_0$ counts the number of nonzero elements of x.

Matrix generalizations of (1.3) and (1.2), respectively, are the so-called matrix rank minimization problem

$$\min_{X\in\mathbb{R}^{m\times n}} \operatorname{rank}(X) \quad \text{s.t.} \quad \mathcal{A}(X) = d, \tag{1.4}$$

[*] Corresponding author.

[†] Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027. Email: {bh2359,sm2756,goldfarb}@columbia.edu. Research supported in part by NSF Grants DMS 06-06712 and DMS 10-16571, ONR Grants N00014-03-0514 and N00014-08-1-1118, and DOE Grants DE-FG01-92ER-25126 and DE-FG02-08ER-25856.

and its convex relaxation, the nuclear norm minimization problem:

$$\min_{X \in \mathbb{R}^{m\times n}} \|X\|_* \quad \text{s.t.} \quad \mathcal{A}(X) = d, \tag{1.5}$$

where $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^p$ is a linear operator, $d \in \mathbb{R}^p$, and $\|X\|_*$ (the nuclear norm of X) is defined as the sum of singular values of matrix X. A special case of (1.4) is the matrix completion problem:

$$\min_{X \in \mathbb{R}^{m\times n}} \operatorname{rank}(X) \quad \text{s.t.} \quad X_{ij} = M_{ij}, \forall (i,j) \in \Omega, \tag{1.6}$$

whose convex relaxation is given by:

$$\min_{X \in \mathbb{R}^{m\times n}} \|X\|_* \quad \text{s.t.} \quad X_{ij} = M_{ij}, \forall (i,j) \in \Omega. \tag{1.7}$$

The matrix completion problem has a lot of interesting applications in online recommendation systems, collaborative filtering [35, 36], etc., including the famous Netflix problem [34]. It has been proved that under certain conditions, the solutions of the NP-hard problems (1.4) and (1.6) are given respectively by solving their convex relaxations (1.5) and (1.7), with high probability (see, e.g., [31, 8, 10, 30, 15]).

The linearized Bregman (LB) method was proposed in [42] to solve the basis pursuit problem (1.2). The method was derived by linearizing the quadratic penalty term in the augmented Lagrangian function that is minimized on each iteration of the so-called Bregman method introduced in [27] while adding a prox term to it. The linearized Bregman method was further analyzed in [5, 7, 41] and applied to solve the matrix completion problem (1.7) in [5].

Throughout of this paper, we will sometimes focus our analysis on the basis pursuit problem (1.2). However, all of the analysis and results can be easily extended to (1.5) and (1.7). The linearized Bregman method depends on a single parameter $\mu > 0$ and, as the analysis in [5, 7] shows, actually solves the problem

$$\min_{x \in \mathbb{R}^n} g_\mu(x) := \|x\|_1 + \frac{1}{2\mu}\|x\|_2^2, \quad \text{s.t.} \quad Ax = b, \tag{1.8}$$

rather than the problem (1.2). Recently it was shown in [41] that the solution to (1.8) is also a solution to problem (1.2) as long as μ is chosen large enough. Furthermore, it was shown in [41] that the linearized Bregman method can be viewed as a gradient descent method applied to the Lagrangian dual of problem (1.8). This dual problem is an unconstrained optimization problem of the form

$$\min_{y \in \mathbb{R}^m} \quad G_\mu(y), \tag{1.9}$$

where the objective function $G_\mu(y)$ is differentiable since $g_\mu(x)$ is strictly convex (see, e.g., [32]). Motivated by this result, some techniques for speeding up the classical gradient descent method applied to this dual problem such as taking Barzilai-Borwein (BB) steps [1], and incorporating it into a limited memory BFGS (L-BFGS) method [18], were proposed in [41]. Numerical results on the basis pursuit problem (1.2) reported in [41] show that the performance of the linearized Bregman method can be greatly improved by using these techniques.

Our starting point is also motivated by the equivalence between applying the linearized Bregman method to (1.2) and solving the Lagrangian dual problem (1.9) by the gradient descent method. Since the gradient of $G_\mu(y)$ can be shown to be Lipschitz continuous, it is well-known that the classical gradient descent method with a properly chosen step size will obtain an ϵ-optimal solution to (1.9) (i.e., an approximate solution y^k such that $G_\mu(y^k) - G_\mu(y^*) \leq \epsilon$) in $O(1/\epsilon)$ iterations. In [23], Nesterov proposed a technique for accelerating the gradient descent method for solving problem of the form (1.9) (see, also, [24]), and proved that using this accelerated method, the number of iterations needed to obtain an ϵ-optimal solution is reduced to $O(1/\sqrt{\epsilon})$ with a negligible change in the work required at each iteration. Nesterov also proved that the $O(1/\sqrt{\epsilon})$ complexity bound is the best bound that one can get if one uses only the first-order information. Based on the above discussion, we propose an accelerated linearized Bregman (ALB) method for solving (1.8) which is equivalent to an accelerated gradient descent method for solving the Lagrangian dual (1.9) of (1.8). As a by-product, we show that the basic and the accelerated linearized Bregman methods require $O(1/\epsilon)$ and $O(1/\sqrt{\epsilon})$ iterations, respectively, to obtain an ϵ-optimal solution with respect to the Lagrangian for (1.8).

The rest of this paper is organized as follows. In Section 2 we describe the original Bregman iterative method, as well as the linearized Bregman method. We motivate the methods and state some previously obtained theoretical results that establish the equivalence between the LB method and a gradient descent method for the dual of problem (1.8). We present our accelerated linearized Bregman method in Section 3. We also provide a theoretical foundation for the accelerated algorithm and prove complexity results for it and the unaccelerated method. In Section 4, we describe how the LB and ALB methods can be extended to basis pursuit problems that include additional convex constraints. In Section 5, we report preliminary numerical results, on several compressed sensing basis pursuit and matrix completion problems. These numerical results show that our accelerated linearized Bregman method significantly outperforms the basic linearized Bregman method. We make some conclusions in Section 6.

2. Bregman and Linearized Bregman Methods.

The Bregman method was introduced to the image processing community by Osher et al. in [27] for solving the total-variation (TV) based image restoration problems. The Bregman distance [4] with respect to convex function $J(\cdot)$ between points u and v is defined as

$$D_J^p(u,v) := J(u) - J(v) - \langle p, u - v\rangle, \tag{2.1}$$

where $p \in \partial J(v)$, the subdifferential of J at v. The Bregman method for solving (1.1) is given below as Algorithm 1. Note that the updating formula for p^k (Step 4 in Algorithm 1) is based on the optimality conditions of Step 3 in Algorithm 1:

$$0 \in \partial J(x^{k+1}) - p^k + A^\top(Ax^{k+1} - b).$$

This leads to

$$p^{k+1} = p^k - A^\top(Ax^{k+1} - b).$$

It was shown in [27, 42] that the Bregman method (Algorithm 1) converges to a solution of (1.1) in a finite number of steps.

It is worth noting that for solving (1.1), the Bregman method is equivalent to the augmented Lagrangian method [17, 29, 33] in the following sense.

Theorem 2.1. *The sequences $\{x^k\}$ generated by Algorithm 1 and by the augmented Lagrangian method,*

Algorithm 1 Original Bregman Iterative Method

```
Input: $x^0 = p^0 = 0$.
for $k = 0, 1, \cdots$ do
    $x^{k+1} = \arg\min_x D_J^{p^k}(x, x^k) + \frac{1}{2}\|Ax - b\|^2$;
    $p^{k+1} = p^k - A^\top(Ax^{k+1} - b)$;
end for
```

which computes for $k = 0, 1, \cdots$

$$\left\{\begin{array}{lll} x^{k+1} & := & \arg\min_x J(x) - \langle \lambda^k, Ax - b\rangle + \frac{1}{2}\|Ax - b\|^2 \\ \lambda^{k+1} & := & \lambda^k - (Ax^{k+1} - b) \end{array}\right. \tag{2.2}$$

starting from $\lambda^0 = 0$ are exactly the same.

Proof. From Step 4 of Algorithm 1 and the fact that $p^0 = 0$, it follows that $p^k = -\sum_{j=1}^k A^\top(Ax^j - b)$. From the second equation in (2.2) and using $\lambda^0 = 0$, we get $\lambda^k = -\sum_{j=1}^k (Ax^j - b)$. Thus, $p^k = A^\top\lambda^k$ for all k. Hence it is easy to see that Step 3 of Algorithm 1 is exactly the same as the first equation in (2.2) and that the x^{k+1} computed in Algorithm 1 and (2.2) are exactly the same. Therefore, the sequences $\{x^k\}$ generated by both algorithms are exactly the same. □

Note that for $J(x) := \alpha\|x\|_1$, Step 3 of Algorithm 1 reduces to an ℓ_1-regularized problem:

$$\min_x \quad \alpha\|x\|_1 - \langle p^k, x\rangle + \frac{1}{2}\|Ax - b\|^2. \tag{2.3}$$

Although there are many algorithms for solving the subproblem (2.3) such as FPC [16], SPGL1 [39], FISTA [2] etc., it often takes them many iterations to do so. The linearized Bregman method was proposed in [42], and used in [28, 7, 6] to overcome this difficulty. The linearized Bregman method replaces the quadratic term $\frac{1}{2}\|Ax - b\|^2$ in the objective function that is minimized in Step 3 of Algorithm 1 by its linearization $\langle A^\top(Ax^k - b), x\rangle$ plus a proximal term $\frac{1}{2\mu}\|x - x^k\|^2$. Consequently the updating formula for p^k is changed since the optimality conditions for this minimization step become:

$$0 \in \partial J(x^{k+1}) - p^k + A^\top(Ax^k - b) + \frac{1}{\mu}(x^{k+1} - x^k).$$

In Algorithm 2 below we present a slightly generalized version of the original linearized Bregman method that includes an additional parameter τ that corresponds to the length of a gradient step in a dual problem.

Algorithm 2 Linearized Bregman Method

```
Input: $x^0 = p^0 = 0$, $\mu > 0$ and $\tau > 0$.
for $k = 0, 1, \cdots$ do
    $x^{k+1} = \arg\min_x D_J^{p^k}(x, x^k) + \tau\langle A^\top(Ax^k - b), x\rangle + \frac{1}{2\mu}\|x - x^k\|^2$;
    $p^{k+1} = p^k - \tau A^\top(Ax^k - b) - \frac{1}{\mu}(x^{k+1} - x^k)$;
end for
```

In [41], it is shown that when $\mu\|A\|^2 < 2$, where $\|A\|$ denotes the largest singular value of A, the iterates of the linearized Bregman method (Algorithm 2 with $\tau = 1$) converge to the solution of the following regularized version of problem (1.1):

$$\min_x \quad J(x) + \frac{1}{2\mu}\|x\|^2 \quad \text{s.t.} \quad Ax = b. \tag{2.4}$$

We prove in Theorem 2.3 below an analogous result for Algorithm 2 for a range of values of τ. However, we first prove, as in [41], that the linearized Bregman method (Algorithm 2) is equivalent to a gradient descent method

$$y^{k+1} := y^k - \tau \nabla G_\mu(y^k) \tag{2.5}$$

applied to the Lagrangian dual

$$\max_y \min_w \{J(w) + \frac{1}{2\mu}\|w\|^2 - \langle y, Aw - b\rangle\}$$

of (2.4), which we express as the following equivalent minimization problem:

$$\min_y \quad G_\mu(y) := -\{J(w^*) + \frac{1}{2\mu}\|w^*\|^2 - \langle y, Aw^* - b\rangle\}, \tag{2.6}$$

where

$$w^* := \arg\min_w \{J(w) + \frac{1}{2\mu}\|w\|^2 - \langle y, Aw - b\rangle\}.$$

To show that $G_\mu(y)$ is continuously differentiable, we rewrite $G_\mu(y)$ as

$$G_\mu(y) = -\Phi_\mu(\mu A^\top y) + \frac{\mu}{2}\|A^\top y\|^2 - b^\top y,$$

where

$$\Phi_\mu(v) \equiv \min_w \{J(w) + \frac{1}{2\mu}\|w - v\|^2\}$$

is strictly convex and continuously differentiable with gradient $\nabla \Phi_\mu(v) = \frac{v - \hat{w}}{\mu}$, and $\hat{w} = \arg\min_w \{J(w) + \frac{1}{2\mu}\|w - v\|^2\}$ (e.g., see Proposition 4.1 in [3]). From this it follows that $\nabla G_\mu(y) = Aw^* - b$. Hence the gradient method (2.5) corresponds to Algorithm 3 below.

Algorithm 3 Linearized Bregman Method (Equivalent Form)

```
1: Input: $\mu > 0$, $\tau > 0$ and $y^0 = \tau b$.
2: for $k = 0, 1, \cdots$ do
3:     $w^{k+1} := \arg\min_w \{J(w) + \frac{1}{2\mu}\|w\|^2 - \langle y^k, Aw - b\rangle\}$;
4:     $y^{k+1} := y^k - \tau(Aw^{k+1} - b)$.;
5: end for
```

Lemma 2.2 and Theorem 2.3 below generalize Theorem 2.1 in [41] by allowing a step length choice in the gradient step (2.5) and show that Algorithms 2 and 3 are equivalent. Our proof closely follows the proof of Theorem 2.1 in [41].

LEMMA 2.2. *x^{k+1} computed by Algorithm 2 equals w^{k+1} computed by Algorithm 3 if and only if*

$$A^\top y^k = p^k - \tau A^\top (Ax^k - b) + \frac{1}{\mu} x^k. \tag{2.7}$$

Proof. By comparing Step 3 in Algorithms 2 and 3, it is obvious that w^{k+1} is equal to x^{k+1} if and only if (2.7) holds. □

Theorem 2.3. *The sequences $\{x^k\}$ and $\{w^k\}$ generated by Algorithms 2 and 3 are the same.*

Proof. We prove by induction that equation (2.7) holds for all $k \geq 0$. Note that (2.7) holds for $k = 0$ since $p^0 = x^0 = 0$ and $y^0 = \tau b$. Now let us assume that (2.7) holds for all $0 \leq k \leq n-1$; thus by Lemma 2.2 $w^{k+1} = x^{k+1}$ for all $0 \leq k \leq n-1$. By iterating Step 4 in Algorithm 3 we get

$$y^n = y^{n-1} - \tau(Aw^n - b) = -\sum_{j=0}^{n} \tau(Ax^j - b). \tag{2.8}$$

By iterating Step 4 in Algorithm 2 we get

$$p^n = -\sum_{j=0}^{n-1} \tau A^\top(Ax^j - b) - \frac{1}{\mu}x^n,$$

which implies that

$$p^n - \tau A^\top(Ax^n - b) + \frac{1}{\mu}x^n = -\sum_{j=0}^{k} \tau A^\top(Ax^j - b) = A^\top y^n,$$

where the last equality follows from (2.8); thus by induction (2.7) holds for all $k \geq 0$, which implies by Lemma 2.2 that $x^k = w^k$ for all $k \geq 0$. □

Before analyzing Algorithms 2 and 3, we note that by defining $v^k = A^\top y^k$ and algebraically manipulating the last two terms in the objective function in Step 3 in Algorithm 3, Steps 3 and 4 in that algorithm can be replaced by

$$\left\{ \begin{array}{lll} w^{k+1} & := & \arg\min_w J(w) + \frac{1}{2\mu}\|w - \mu v^k\|^2 \\ v^{k+1} & := & v^k - \tau A^\top(Aw^{k+1} - b) \end{array} \right. \tag{2.9}$$

if we set $v^0 = \tau A^\top b$. Because Algorithms 2 and 3 are equivalent, convergence results for the gradient descent method can be applied to both of them. Thus we have the following convergence result.

Theorem 2.4. *Let $J(w) \equiv \|w\|_1$. Then $G_\mu(y)$ in the dual problem* (2.6) *is continuously differentiable and its gradient is Lipschitz continuous with the Lipschitz constant $L \leq \mu\|A\|^2$. Consequently, if the step length $\tau < \frac{2}{\mu\|A\|^2}$, the sequences $\{x^k\}$ and $\{w^k\}$ generated by Algorithms 2 and 3 converge to the optimal solution of* (2.4).

Proof. When $J(x) = \|x\|_1$, w^{k+1} in (2.9) reduces to

$$w^{k+1} = \mu \cdot \text{shrink}(v^k, 1),$$

where the ℓ_1 shrinkage operator is defined as

$$\text{shrink}(z, \alpha) := \text{sgn}(z) \circ \max\{|z| - \alpha, 0\}, \forall z \in \mathbb{R}^n, \alpha > 0. \tag{2.10}$$

$G_\mu(y)$ is continuously differentiable since $g_\mu(x)$ is strictly convex. Since for any point y, $\nabla G_\mu(y) = Aw - b$, where $w = \mu \cdot \text{shrink}(A^\top y, 1)$, it follows from the fact that the shrinkage operator is non-expansive, i.e.,

$$\|\text{shrink}(s, \alpha) - \text{shrink}(t, \alpha)\| \leq \|s - t\|, \ \forall s, t, \alpha$$

that

$$
\begin{aligned}
\|\nabla G_\mu(y^1) - \nabla G_\mu(y^2)\| &= \|\mu \cdot A \cdot \mathrm{shrink}(A^\top y^1, 1) - \mu \cdot A \cdot \mathrm{shrink}(A^\top y^2, 1)\| \\
&\leq \mu \cdot \|A\| \cdot \|A^\top (y^1 - y^2)\| \\
&\leq \mu \|A\|^2 \|y^1 - y^2\|,
\end{aligned}
$$

for any two points y^1 and y^2. Thus the Lipschitz constant L of $\nabla G_\mu(\cdot)$ is bounded above by $\mu\|A\|^2$.

When $\tau < \frac{2}{\mu\|A\|^2}$, we have $\tau L < 2$ and thus $|1 - \tau L| < 1$. It then follows that the gradient descent method $y^{k+1} = y^k - \tau \nabla G_\mu(y^k)$ converges and therefore Algorithms 2 and 3 converge to x^*_μ, the optimal solution of (2.4). □

Before developing an accelerated version of the LB algorithm in the next section. We would like to comment on the similarities and differences between the LB method and Nesterov's composite gradient method [26] and the ISTA method [2] applied to problem (1.1) and related problems. The latter algorithms iterate Step 3 in the LB method (Algorithm 2) with $p^k = 0$, and never compute or update the subgradient vector p^k. More importantly, their methods solve the unconstrained problem

$$
\min_{x \in \mathbb{R}^n} \quad \|x\|_1 + \frac{1}{2\mu}\|Ax - b\|^2.
$$

Hence, while these methods and the LB method both linearize the quadratic term $\|Ax - b\|^2$ while handling the nonsmooth term $\|x\|_1$ directly, they are very different.

Similar remarks apply to the accelerated LB method presented in the next section and fast versions of ISTA and Nesterov's composite gradient method.

3. The Accelerated Linearized Bregman Algorithm.

Based on Theorem 2.3, i.e., the equivalence between the linearized Bregman method and the gradient descent method, we can accelerate the linearized Bregman method by techniques used to accelerate the classical gradient descent method. In [41], Yin considered several techniques such as line search, BB step and L-BFGS, to accelerate the linearized Bregman method. Here we consider the acceleration technique proposed by Nesterov in [23, 24]. This technique accelerates the classical gradient descent method in the sense that it reduces the iteration complexity significantly without increasing the per-iteration computational effort. For the unconstrained minimization problem (1.9), Nesterov's accelerated gradient method replaces the gradient descent method (2.5) by the following iterative scheme:

$$
\left\{
\begin{array}{lcl}
x^{k+1} & := & y^k - \tau \nabla G_\mu(y^k) \\
y^{k+1} & := & \alpha_k x^{k+1} + (1 - \alpha_k) x^k,
\end{array}
\right.
\tag{3.1}
$$

where the scalars α_k are specially chosen weighting parameters. A typical choice for α_k is $\alpha_k = \frac{3}{k+2}$. If τ is chosen so that $\tau \leq 1/L$, where L is the Lipschitz constant for $\nabla G_\mu(\cdot)$, Nesterov's accelerated gradient method (3.1) obtains an ϵ-optimal solution of (1.9) in $O(1/\sqrt{\epsilon})$ iterations, while the classical gradient method (2.5) takes $O(1/\epsilon)$ iterations. Moreover, the per-iteration complexities of (2.5) and (3.1) are almost the same since computing the gradient $\nabla G_\mu(\cdot)$ usually dominates the computational cost in each iteration. Nesterov's acceleration technique has been studied and extended by many others for nonsmooth minimization problems and variational inequalities, e.g., see [25, 26, 2, 38, 22, 12, 13, 14].

Our accelerated linearized Bregman method is given below as Algorithm 4. The main difference between it and the basic linearized Bregman method (Algorithm 2) is that the latter uses the previous iterate x^k

and subgradient p^k to compute the new iterate x^{k+1}, while Algorithm 4 uses extrapolations $\tilde{x}^k$ and $\tilde{p}^k$ that are computed as linear combinations of the two previous iterates and subgradients, respectively. Carefully choosing the sequence of weighting parameters $\{\alpha_k\}$ guarantees an improved rate of convergence.

Algorithm 4 Accelerated Linearized Bregman Method

```
1: Input: $x^0 = \tilde{x}^0 = \tilde{p}^0 = p^0 = 0$, $\mu > 0$, $\tau > 0$.
2: for $k = 0, 1, \cdots$ do
3:     $x^{k+1} = \arg\min_x D_J^{\tilde{p}^k}(x, \tilde{x}^k) + \tau\langle A^\top(A\tilde{x}^k - b), x\rangle + \frac{1}{2\mu}\|x - \tilde{x}^k\|^2$;
4:     $p^{k+1} = \tilde{p}^k - \tau A^\top(A\tilde{x}^k - b) - \frac{1}{\mu}(x^{k+1} - \tilde{x}^k)$;
5:     $\tilde{x}^{k+1} = \alpha_k x^{k+1} + (1 - \alpha_k)x^k$;
6:     $\tilde{p}^{k+1} = \alpha_k p^{k+1} + (1 - \alpha_k)p^k$.
7: end for
```

In the following, we first establish the equivalence between the accelerated linearized Bregman method and the corresponding accelerated gradient descent method (3.1), which we give explicitly as (3.2) below applied to the dual problem (2.6). Based on this, we then present complexity results for both basic and accelerated linearized Bregman methods. Not surprisingly, the accelerated linearized Bregman method improves the iteration complexity from $O(1/\epsilon)$ to $O(1/\sqrt{\epsilon})$.

Theorem 3.1. *The accelerated linearized Bregman method (Algorithm 4) is equivalent to the accelerated dual gradient descent method* (3.2) *starting from $\tilde{y}^0 = y^0 = \tau b$:*

$$\left\{\begin{array}{lcl} w^{k+1} & := & \arg\min J(w) + \frac{1}{2\mu}\|w\|^2 - \langle \tilde{y}^k, Aw - b\rangle \\ y^{k+1} & := & \tilde{y}^k - \tau(Aw^{k+1} - b) \\ \tilde{y}^{k+1} & := & \alpha_k y^{k+1} + (1 - \alpha_k)y^k. \end{array}\right. \tag{3.2}$$

More specifically, the sequence $\{x^k\}$ generated by Algorithm 4 is exactly the same as the sequence $\{w^k\}$ generated by (3.2).

Proof. Note that the Step 3 of Algorithm 4 is equivalent to

$$x^{k+1} := \arg\min J(x) - \langle \tilde{p}^k, x\rangle + \tau\langle A^\top(A\tilde{x}^k - b), x\rangle + \frac{1}{2\mu}\|x - \tilde{x}^k\|^2. \tag{3.3}$$

Comparing (3.3) with the first equation in (3.2), it is easy to see that $x^{k+1} = w^{k+1}$ if and only if

$$A^\top\tilde{y}^k = \tilde{p}^k + \tau A^\top(b - A\tilde{x}^k) + \frac{1}{\mu}\tilde{x}^k. \tag{3.4}$$

We will prove (3.4) in the following by induction. Note that (3.4) holds for $k = 0$ since $\tilde{y}^0 = \tau b$ and $\tilde{x}^0 = \tilde{p}^0 = 0$. As a result, we have $x^1 = w^1$. By defining $w^0 = 0$, we also have $x^0 = w^0$,

$$A^\top\tilde{y}^1 = A^\top(\alpha_0 y^1 + (1 - \alpha_0)A^\top y^0) = \alpha_0 A^\top\tilde{y}^0 + \alpha_0\tau A^\top(b - Aw^1) + (1 - \alpha_0)A^\top y^0. \tag{3.5}$$

On the other hand,

$$p^1 = \tilde{p}^0 + \tau A^\top(b - A\tilde{x}^0) - \frac{1}{\mu}(x^1 - \tilde{x}^0) = A^\top\tilde{y}^0 - \frac{1}{\mu}x^1, \tag{3.6}$$

where for the second equality we used (3.4) for $k = 0$. Expressing $\tilde{p}^1$ and $\tilde{x}^1$ in terms of their affine combinations of p^1, p^0, x^1 and x^0, then substituting for p^1 using (3.6) and using the fact that $x^0 = p^0 = 0$, and finally using $\tilde{y}^0 = \tau b$ and (3.5), we obtain,

$$\begin{aligned}\tilde{p}^1 + \tau A^\top(b - A\tilde{x}^1) + \frac{1}{\mu}\tilde{x}^1 &= \alpha_0 p^1 + (1-\alpha_0)p^0 + \alpha_0\tau A^\top(b - Ax^1) + (1-\alpha_0)\tau A^\top(b - Ax^0) + \frac{1}{\mu}(\alpha_0 x^1 + (1-\alpha_0)x^0)\\ &= \alpha_0(A^\top\tilde{y}^0 - \frac{1}{\mu}x^1) + \alpha_0\tau A^\top(b - Ax^1) + (1-\alpha_0)\tau A^\top b + \frac{1}{\mu}\alpha_0 x^1\\ &= \alpha_0 A^\top\tilde{y}^0 + \alpha_0\tau A^\top(b - Ax^1) + (1-\alpha_0)A^\top y^0\\ &= \alpha_0 A^\top\tilde{y}^0 + \alpha_0\tau A^\top(b - Aw^1) + (1-\alpha_0)A^\top y^0\\ &= A^\top\tilde{y}^1.\end{aligned}$$

Thus we proved that (3.4) holds for $k = 1$. Now let us assume that (3.4) holds for $0 \le k \le n-1$, which implies $x^k = w^k, \forall 0 \le k \le n$ since $x^0 = w^0$. We will prove that (3.4) holds for $k = n$.

First, note that

$$p^n = \tilde{p}^{n-1} + \tau A^\top(b - A\tilde{x}^{n-1}) - \frac{1}{\mu}(x^n - \tilde{x}^{n-1}) = A^\top\tilde{y}^{n-1} - \frac{1}{\mu}x^n, \tag{3.7}$$

where the first equality is from Step 4 of Algorithm 4 and the second equality is from (3.4) for $k = n-1$. From Step 6 of Algorithm 4 and (3.7), we have

$$\begin{aligned}\tilde{p}^n &= \alpha_{n-1}p^n + (1-\alpha_{n-1})p^{n-1}\\ &= \alpha_{n-1}(A^\top\tilde{y}^{n-1} - \tfrac{1}{\mu}x^n) + (1-\alpha_{n-1})(A^\top\tilde{y}^{n-2} - \tfrac{1}{\mu}x^{n-1})\\ &= \alpha_{n-1}A^\top\tilde{y}^{n-1} + (1-\alpha_{n-1})A^\top\tilde{y}^{n-2} - \tfrac{1}{\mu}\tilde{x}^n,\end{aligned} \tag{3.8}$$

where the last equality uses Step 5 of Algorithm 4. On the other hand, from (3.2) we have

$$\begin{aligned}A^\top\tilde{y}^n &= A^\top(\alpha_{n-1}y^n + (1-\alpha_{n-1})y^{n-1})\\ &= \alpha_{n-1}A^\top(\tilde{y}^{n-1} + \tau(b - Aw^n)) + (1-\alpha_{n-1})A^\top(\tilde{y}^{n-2} + \tau(b - Aw^{n-1}))\\ &= \alpha_{n-1}A^\top\tilde{y}^{n-1} + (1-\alpha_{n-1})A^\top\tilde{y}^{n-2} + \tau A^\top[b - A(\alpha_{n-1}x^n + (1-\alpha_{n-1})x^{n-1})]\\ &= \alpha_{n-1}A^\top\tilde{y}^{n-1} + (1-\alpha_{n-1})A^\top\tilde{y}^{n-2} + \tau A^\top(b - A\tilde{x}^n),\end{aligned} \tag{3.9}$$

where the third equality is from $w^n = x^n$ and $w^{n-1} = x^{n-1}$, the last equality is from Step 5 of Algorithm 4. Combining (3.8) and (3.9) we get that (3.4) holds for $k = n$. □

Like the linearized Bregman, we can also use a simpler implementation for accelerated linearized Bregman method in which the main computation at each step is a proximal minimization. Specifically, (3.2) is equivalent to the following three steps.

$$\begin{cases} w^{k+1} &:= \arg\min J(w) + \frac{1}{2\mu}\|w - \mu\tilde{v}^k\|^2\\ v^{k+1} &:= \tilde{v}^k - \tau A^\top(Aw^{k+1} - b)\\ \tilde{v}^{k+1} &:= \alpha_k v^{k+1} + (1-\alpha_k)v^k\end{cases} \tag{3.10}$$

As before this follows from letting $v^k = A^\top y^k$ and $\tilde{v}^k = A^\top\tilde{y}^k$ and completing the square in the objective function in the first equation of (3.2).

Next we prove iteration complexity bounds for both basic and accelerated linearized Bregman algorithms. Since these algorithms are standard gradient descent methods applied to the Lagrangian dual function and these results have been well established, our proofs will be quite brief.

THEOREM 3.2. *Let the sequence $\{x^k\}$ be generated by the linearized Bregman method (Algorithm 2) and (x^*, y^*) be the pair of optimal primal and dual solutions for Problem (2.4). Let $\{y^k\}$ be the sequence generated by Algorithm 3 and suppose the step length $\tau \leq \frac{1}{L}$, where L is the Lipschitz constant for $\nabla G_\mu(y)$. Then for the Lagrangian function*

$$\mathcal{L}_\mu(x,y) = J(x) + \frac{1}{2\mu}\|x\|^2 - \langle y, Ax-b\rangle, \tag{3.11}$$

we have

$$\mathcal{L}_\mu(x^*,y^*) - \mathcal{L}_\mu(x^{k+1},y^k) \leq \frac{\|y^*-y^0\|^2}{2\tau k}. \tag{3.12}$$

Thus, if we further have $\tau \geq \beta/L$, where $0 < \beta \leq 1$, then (x^{k+1}, y^k) is an ϵ-optimal solution to Problem (2.4) with respect to the Lagrangian function if $k \geq \lceil C/\epsilon \rceil$, where $C := \frac{L\|y^*-y^0\|^2}{2\beta}$.

Proof. From (2.6) we get

$$G_\mu(y^k) = -\mathcal{L}_\mu(x^{k+1}, y^k). \tag{3.13}$$

By using the convexity of function $G_\mu(\cdot)$ and the Lipschitz continuity of the gradient $\nabla G_\mu(\cdot)$, we get for any y,

$$\begin{aligned}
G_\mu(y^k) - G_\mu(y) &\leq G_\mu(y^{k-1}) + \langle \nabla G_\mu(y^{k-1}), y^k - y^{k-1}\rangle + \tfrac{L}{2}\|y^k - y^{k-1}\|^2 - G_\mu(y) \\
&\leq G_\mu(y^{k-1}) + \langle \nabla G_\mu(y^{k-1}), y^k - y^{k-1}\rangle + \tfrac{1}{2\tau}\|y^k - y^{k-1}\|^2 - G_\mu(y) \\
&\leq \langle \nabla G_\mu(y^{k-1}), y^{k-1} - y\rangle + \langle \nabla G_\mu(y^{k-1}), y^k - y^{k-1}\rangle + \tfrac{1}{2\tau}\|y^k - y^{k-1}\|^2 \\
&= \langle \nabla G_\mu(y^{k-1}), y^k - y\rangle + \tfrac{1}{2\tau}\|y^k - y^{k-1}\|^2 \\
&= \tfrac{1}{\tau}\langle y^{k-1} - y^k, y^k - y\rangle + \tfrac{1}{2\tau}\|y^k - y^{k-1}\|^2 \\
&\leq \tfrac{1}{2\tau}(\|y - y^{k-1}\|^2 - \|y - y^k\|^2).
\end{aligned} \tag{3.14}$$

Setting $y = y^{k-1}$ in (3.14), we obtain $G_\mu(y^k) \leq G_\mu(y^{k-1})$ and thus the sequence $\{G_\mu(y^k)\}$ is non-increasing. Moreover, summing (3.14) over $k = 1, 2, \ldots, n$ with $y = y^*$ yields

$$n(G_\mu(y^n) - G_\mu(y^*)) \leq \sum_{k=1}^{n}(G_\mu(y^k) - G_\mu(y^*)) \leq \frac{1}{2\tau}(\|y^* - y^0\|^2 - \|y^* - y^n\|^2) \leq \frac{1}{2\tau}\|y^* - y^0\|^2,$$

and this implies (3.12). □

Before we analyze the iteration complexity of the accelerated linearized Bregman method, we introduce a lemma from [38] that we will use in our analysis.

LEMMA 3.3 (Property 1 in [38]). *For any proper lower semicontinuous function $\psi : \mathbb{R}^n \to (-\infty, +\infty]$ and any $z \in \mathbb{R}^n$, if*

$$z_+ = \arg\min_x \{\psi(x) + \frac{1}{2}\|x - z\|^2\},$$

then

$$\psi(x) + \frac{1}{2}\|x - z\|^2 \geq \psi(z_+) + \frac{1}{2}\|z_+ - z\|^2 + \frac{1}{2}\|x - z_+\|^2, \quad \forall x \in \mathbb{R}^n.$$

The following theorem gives an iteration-complexity result for the accelerated linearized Bregman method. Our proof of this theorem closely follows the proof of Proposition 2 in [38].

THEOREM 3.4. *Let the sequence $\{x^k\}$ be generated by accelerated linearized Bregman method (Algorithm 4) and (x^*, y^*) be the optimal primal and dual variable for Problem (2.4). Let $\{\alpha_k\}$ be chosen as*

$$\alpha_{k-1} = 1 + \theta_k(\theta_{k-1}^{-1} - 1), \tag{3.15}$$

where

$$\theta_{-1} := 1, \text{ and } \theta_k = \frac{2}{k+2}, \forall k \geq 0. \tag{3.16}$$

Let the sequence $\{y^k\}$ be defined as in (3.2) and the step length $\tau \leq \frac{1}{L}$, where L is the Lipschitz constant of $\nabla G_\mu(y)$ and $G_\mu(\cdot)$ is defined by (3.13). We have

$$G_\mu(y^k) - G_\mu(y^*) \leq \frac{2\|y^* - y^0\|^2}{\tau k^2}. \tag{3.17}$$

Thus, if we further have $\tau \geq \beta/L$, where $0 < \beta \leq 1$, then (x^{k+1}, y^k) is an ϵ-optimal solution to Problem (2.4) with respect to the Lagrangian function (3.11) if $k \geq \lceil\sqrt{C/\epsilon}\rceil$, where $C := \frac{2L\|y^*-y^0\|^2}{\beta}$.

Proof. Let

$$z^k = y^{k-1} + \theta_{k-1}^{-1}(y^k - y^{k-1}) \tag{3.18}$$

and denote the linearization of $G_\mu(y)$ as

$$l_{G_\mu}(x; y) := G_\mu(y) + \langle \nabla G_\mu(y), x - y\rangle \leq G_\mu(x). \tag{3.19}$$

Therefore the second equality in (3.2) is equivalent to

$$\begin{aligned} y^{k+1} := \arg\min_y \quad & G_\mu(\tilde{y}^k) + \langle \nabla G_\mu(\tilde{y}^k), y - \tilde{y}^k\rangle + \frac{1}{2\tau}\|y - \tilde{y}^k\|^2 \\ = \arg\min_y \quad & l_{G_\mu}(y; \tilde{y}^k) + \frac{1}{2\tau}\|y - \tilde{y}^k\|^2. \end{aligned}$$

Define $\hat{y}^k := (1-\theta_k)y^k + \theta_k y^*$, we have

(3.20)

$$\begin{aligned}
G_\mu(y^{k+1}) &\le G_\mu(\tilde{y}^k) + \langle \nabla G_\mu(\tilde{y}^k), y^{k+1} - \tilde{y}^k\rangle + \frac{L}{2}\|y^{k+1} - \tilde{y}^k\|^2 \\
&\le l_{G_\mu}(y^{k+1}; \tilde{y}^k) + \frac{1}{2\tau}\|y^{k+1} - \tilde{y}^k\|^2 \\
&\le l_{G_\mu}(\hat{y}^k; \tilde{y}^k) + \frac{1}{2\tau}\|\hat{y}^k - \tilde{y}^k\|^2 - \frac{1}{2\tau}\|\hat{y}^k - y^{k+1}\|^2 \\
&= l_{G_\mu}((1-\theta_k)y^k + \theta_k y^*; \tilde{y}^k) + \frac{1}{2\tau}\|(1-\theta_k)y^k + \theta_k y^* - \tilde{y}^k\|^2 - \frac{1}{2\tau}\|(1-\theta_k)y^k + \theta_k y^* - y^{k+1}\|^2 \\
&= l_{G_\mu}((1-\theta_k)y^k + \theta_k y^*; \tilde{y}^k) + \frac{\theta_k^2}{2\tau}\|y^* + \theta_k^{-1}(y^k - \tilde{y}^k) - y^k\|^2 - \frac{\theta_k^2}{2\tau}\|y^* + \theta_k^{-1}(y^k - y^{k+1}) - y^k\|^2 \\
&= l_{G_\mu}((1-\theta_k)y^k + \theta_k y^*; \tilde{y}^k) + \frac{\theta_k^2}{2\tau}\|y^* - z^k\|^2 - \frac{\theta_k^2}{2\tau}\|y^* - z^{k+1}\|^2 \\
&= (1-\theta_k)l_{G_\mu}(y^k; \tilde{y}^k) + \theta_k l_{G_\mu}(y^*; \tilde{y}^k) + \frac{\theta_k^2}{2\tau}\|y^* - z^k\|^2 - \frac{\theta_k^2}{2\tau}\|y^* - z^{k+1}\|^2 \\
&\le (1-\theta_k)G_\mu(y^k) + \theta_k G_\mu(y^*) + \frac{\theta_k^2}{2\tau}\|y^* - z^k\|^2 - \frac{\theta_k^2}{2\tau}\|y^* - z^{k+1}\|^2,
\end{aligned}$$

where the second inequality is from (3.19) and $\tau \le 1/L$, the third inequality uses Lemma 3.3 with $\psi(x) := \tau l_{G_\mu}(x; \tilde{y}^k)$, the third equality uses (3.18), (3.2) and (3.15) and the last inequality uses (3.19).

Therefore we get

$$\frac{1}{\theta_k^2}(G_\mu(y^{k+1}) - G_\mu(y^*)) \le \frac{1-\theta_k}{\theta_k^2}(G_\mu(y^k) - G_\mu(y^*)) + \frac{1}{2\tau}\|y - z^k\|^2 - \frac{1}{2\tau}\|y - z^{k+1}\|^2.$$

From (3.16), it is easy to show that $\frac{1-\theta_k}{\theta_k^2} \le \frac{1}{\theta_{k-1}^2}$ for all $k \ge 0$. Thus (3.20) implies that

$$\frac{1-\theta_{k+1}}{\theta_{k+1}^2}(G_\mu(y^{k+1}) - G_\mu(y^*)) \le \frac{1-\theta_k}{\theta_k^2}(G_\mu(y^k) - G_\mu(y^*)) + \frac{1}{2\tau}\|y - z^k\|^2 - \frac{1}{2\tau}\|y - z^{k+1}\|^2. \tag{3.21}$$

Summing (3.21) over $k = 0, 1, \ldots, n-1$, we get

$$\frac{1-\theta_n}{\theta_n^2}(G_\mu(y^n) - G_\mu(y^*)) \le \frac{1}{2\tau}\|y^* - z^0\|^2 = \frac{1}{2\tau}\|y^* - y^0\|^2,$$

which immediately implies (3.17). □

Remark 3.5. *The proof technique and the choice of θ_k used here are suggested in [38] for accelerating the basic algorithm. Other choices of θ_k can be found in [23, 24, 2, 38]. They all work here and give the same order of iteration complexity.*

4. Extension to Problems with Additional Convex Constraints. We now consider extensions of both the LB and ALB methods to problems of the form

$$\min_{x \in X} \quad J(x) \quad \text{s.t} \quad Ax = b, \tag{4.1}$$

where X is a nonempty closed convex set in $\mathbb{R}^n$. It is not clear how to extend the LB and ALB methods (Algorithms 2 and 4) to problem (4.1) since we can no longer rely on the relationship

$$0 \in \partial J(x^{k+1}) - p^k + A^\top(Ax^k - b) + \frac{1}{\mu}(x^{k+1} - x^k)$$

to compute a subgradient $p^{k+1} \in \partial J(x^{k+1})$. Fortunately, the Lagrangian dual gradient versions of these algorithms do not suffer from this difficulty. All that is required to extend them to problem (4.1) is to include the constraint $w \in X$ in the minimization step in these algorithms. Note that the gradient of

$$\hat{\Phi}_\mu(v) = \min_{w \in X}\{J(w) + \frac{1}{2\mu}\|w - v\|^2\}$$

remains the same. Also it is clear that the iteration complexity results given in Theorems 3.2 and 3.4 apply to these algorithms as well.

Being able to apply the LB and ALB methods to problems of the form of (4.1) greatly expands their usefulness. One immediate extension is to compressed sensing problems in which the signal is required to have nonnegative components. Also (4.1) directly includes all linear programs. Applying the LB and ALB to such problems, with the goal of only obtaining approximated optimal solutions, will be the subject of a future paper.

5. Numerical Experiments. In this section, we report some numerical results that demonstrate the effectiveness of the accelerated linearized Bregman algorithm. All numerical experiments were run in MATLAB 7.3.0 on a Dell Precision 670 workstation with an Intel Xeon(TM) 3.4GHZ CPU and 6GB of RAM.

5.1. Numerical Results on Compressed Sensing Problems. In this subsection, we compare the performance of the accelerated linearized Bregman method against the performance of the basic linearized Bregman method on a variety of compressed sensing problems of the form (1.2).

We use three types of sensing matrices $A \in \mathbb{R}^{m\times n}$. Type (i): A is a standard Gaussian matrix generated by the *randn*(m, n) function in MATLAB. Type (ii): A is first generated as a standard Gaussian matrix and then normalized to have unit-norm columns. Type (iii): The elements of A are sampled from a Bernoulli distribution as either $+1$ or -1. We use two types of sparse solutions $x^* \in \mathbb{R}^n$ with sparsity s (i.e., the number of nonzeros in x^*). The positions of the nonzero entries of x^* are selected uniformly at random, and each nonzero value is sampled either from (i) standard Gaussian (the *randn* function in MATLAB) or from (ii) $[-1, 1]$ uniformly at random ($2 * rand - 1$ in MATLAB).

For compressed sensing problems, where $J(x) = \|x\|_1$, the linearized Bregman method reduces to the two-line algorithm:

$$\begin{cases} x^{k+1} & := & \mu \cdot \mathrm{shrink}(v^k, 1) \\ v^{k+1} & := & v^k + \tau A^\top(b - Ax^{k+1}), \end{cases}$$

where the ℓ_1 shrinkage operator is defined in (2.10). Similarly, the accelerated linearized Bregman can be written as:

$$\begin{cases} x^{k+1} & := & \mu \cdot \mathrm{shrink}(\tilde{v}^k, 1) \\ v^{k+1} & := & \tilde{v}^k + \tau A^T(b - Ax^{k+1}) \\ \tilde{v}^{k+1} & := & \alpha_k v^{k+1} + (1 - \alpha_k)v^k. \end{cases}$$

Both algorithms are very simple to program and involve only one Ax and one $A^\top y$ matrix-vector multiplication in each iteration.

We ran both LB and ALB with the *seed* used for generating random number in MATLAB setting as 0. Here we set $n = 2000, m = 0.4 \times n, s = 0.2 \times m, \mu = 5$ for all data sets. We set $\tau = \frac{2}{\mu\|A\|^2}$. We terminated the algorithms when the stopping criterion

$$\|Ax^k - b\|/\|b\| < 10^{-5} \tag{5.1}$$

was satisfied or the number of iterations exceeded 5000. Note that (5.1) was also used in [41]. We report the results in Table 5.1.

Table 5.1
Compare linearized Bregman (LB) with accelerated linearized Bregman (ALB)

Standard Gaussian matrix A		Number of Iterations		Relative error $\|x - x^*\|/\|x^*\|$	
Type of x^*	$n(m = 0.4n, s = 0.2m)$	LB	ALB	LB	ALB
Gaussian	2000	5000+	330	5.1715e-3	1.4646e-5
Uniform	2000	1681	214	2.2042e-5	1.5241e-5
Normalized Gaussian matrix A		Number of Iterations		Relative error $\|x - x^*\|/\|x^*\|$	
Type of x^*	$n(m = 0.4n, s = 0.2m)$	LB	ALB	LB	ALB
Gaussian	2000	2625	234	3.2366e-5	1.2664e-5
Uniform	2000	5000+	292	1.2621e-2	1.5629e-5
Bernoulli +1/-1 matrix A		Number of Iterations		Relative error $\|x - x^*\|/\|x^*\|$	
Type of x^*	$n(m = 0.4n, s = 0.2m)$	LB	ALB	LB	ALB
Gaussian	2000	2314	222	4.2057e-5	1.0812e-5
Uniform	2000	5000+	304	1.6141e-2	1.5732e-5

In Table 5.1, we see that for three out of six problems, LB did not achieve the desired convergence criterion within 5000 iterations, while ALB satisfied this stopping criterion in less than 330 iterations on all six problems. To further demonstrate the significant improvement the ALB achieved over LB, we plot in Figures 5.1, 5.2 and 5.3 the Euclidean norms of the residuals and the relative errors as a function of the iteration number that were obtained by LB and ALB applied to the same data sets. These figures also depict the non-monotonic behavior of the ALB method.

5.2. Numerical Results on Matrix Completion Problems. There are fast implementations of linearized Bregman [5] and other solvers [20, 37, 19, 40] for solving matrix completion problems. We do not compare the linearized Bregman and our accelerated linearized Bregman algorithms with these fast solvers here. Rather our tests are focused only on comparing ALB with LB and verifying that the acceleration actually occurs in practice for matrix completion problems.

The nuclear norm matrix completion problem (1.7) can be rewritten as

$$\min_X \quad \|X\|_* \quad \text{s.t.} \quad \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M), \tag{5.2}$$

where $[\mathcal{P}_\Omega(X)]_{ij} = X_{ij}$ if $(i,j) \in \Omega$ and $[\mathcal{P}_\Omega(X)]_{ij} = 0$ otherwise. When the convex function $J(\cdot)$ is the nuclear norm of matrix X, the Step 3 of Algorithm 2 with inputs X^k, P^k can be reduced to

$$X^{k+1} := \arg\min_{X \in \mathbb{R}^{m\times n}} \mu\|X\|_* + \frac{1}{2}\|X - (X^k - \mu(\tau\mathcal{P}_\Omega(\mathcal{P}_\Omega X^k - \mathcal{P}_\Omega(M)) - P^k))\|_F^2. \tag{5.3}$$

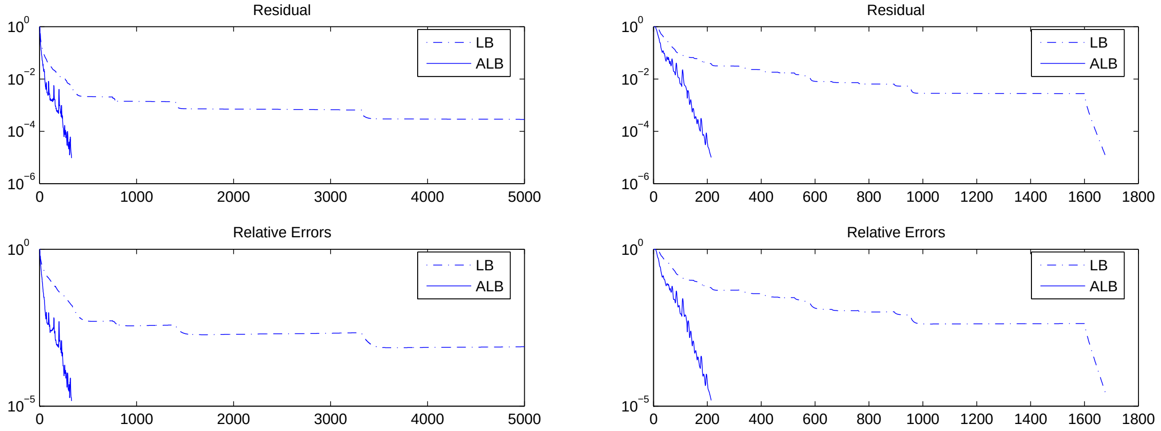

FIG. 5.1. *Gaussian matrix A, Left: Gaussian x^*, Right: Uniform x^**

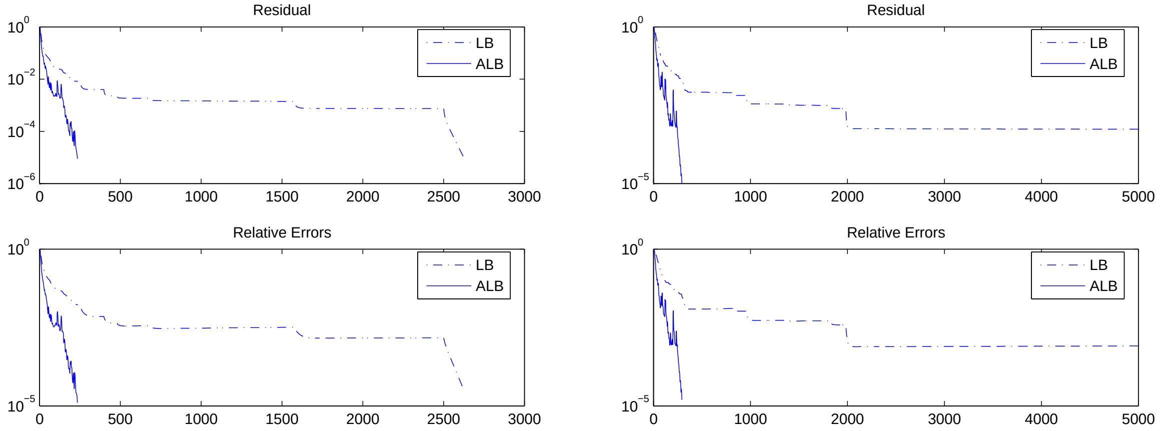

FIG. 5.2. *Normalized Gaussian matrix A, Left: Gaussian x^*, Right: Uniform x^**

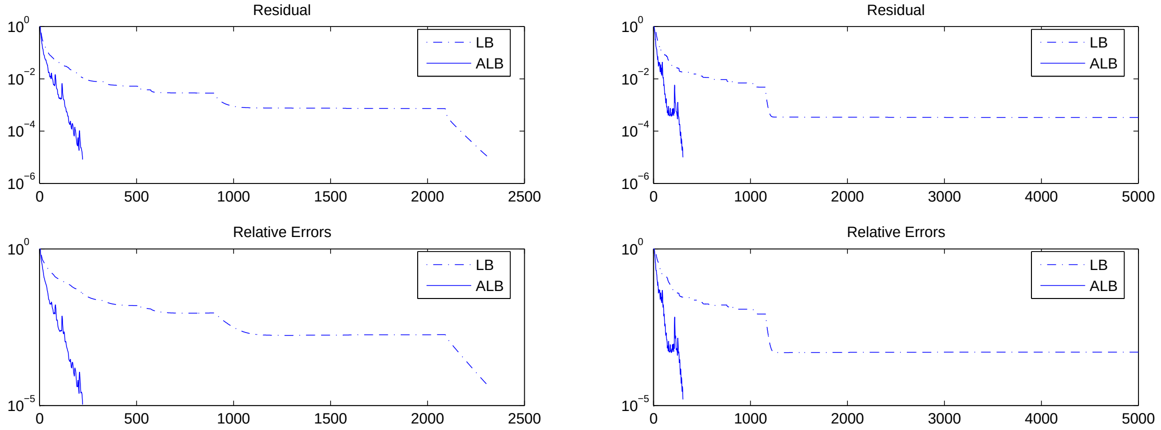

FIG. 5.3. *Bernoulli matrix A, Left: Gaussian x^*, Right: Uniform x^**

It is known (see, e.g., [5, 20]) that (5.3) has the closed-form solution,

$$X^{k+1} = \text{Shrink}(X^k - \mu(\tau\mathcal{P}_\Omega(\mathcal{P}_\Omega X^k - \mathcal{P}_\Omega(M)) - P^k), \mu),$$

where the matrix shrinkage operator is defined as

$$\text{Shrink}(Y, \gamma) := U\text{Diag}(\max(\sigma - \gamma, 0))V^\top,$$

and $Y = U\text{Diag}(\sigma)V^\top$ is the singular value decomposition (SVD) of matrix Y. Thus, a typical iteration of the linearized Bregman method (Algorithm 2), with initial inputs $X^0 = P^0 = 0$, for solving the matrix completion problem (5.2) can be summarized as

$$\left\{\begin{array}{lcl} X^{k+1} & := & \text{Shrink}(X^k - \mu(\tau\mathcal{P}_\Omega(\mathcal{P}_\Omega X^k - \mathcal{P}_\Omega(M)) - P^k), \mu) \\ P^{k+1} & := & P^k - \tau(\mathcal{P}_\Omega X^k - \mathcal{P}_\Omega M) - (X^{k+1} - X^k)/\mu. \end{array}\right. \tag{5.4}$$

Similarly, a typical iteration of the accelerated linearized Bregman method (Algorithm 4), with initial inputs $X^0 = P^0 = \tilde{X}^0 = \tilde{P}^0 = 0$, for solving the matrix completion problem (5.2) can be summarized as

$$\left\{\begin{array}{lcl} X^{k+1} & := & \text{Shrink}(X^k - \mu(\tau\mathcal{P}_\Omega(\mathcal{P}_\Omega X^k - \mathcal{P}_\Omega(M)) - P^k), \mu) \\ P^{k+1} & := & \tilde{P}^k - \tau(\mathcal{P}_\Omega \tilde{X}^k - \mathcal{P}_\Omega M) - (X^{k+1} - \tilde{X}^k)/\mu \\ \tilde{X}^{k+1} & := & \alpha_k X^{k+1} + (1 - \alpha_k)X^k \\ \tilde{P}^{k+1} & := & \alpha_k P^{k+1} + (1 - \alpha_k)P^k, \end{array}\right. \tag{5.5}$$

where the sequence α_k is chosen according to Theorem 3.4.

We compare the performance of LB and ALB on a variety of matrix completion problems. We created matrices $M \in \mathbb{R}^{n\times n}$ with rank r by the following procedure. We first created standard Gaussian matrices $M_L \in \mathbb{R}^{n\times r}$ and $M_R \in \mathbb{R}^{n\times r}$ and then we set $M = M_L M_R^\top$. The locations of the p known entries in M were sampled uniformly, and the values of these p known entries were drawn from an iid Gaussian distribution. The ratio p/n^2 between the number of measurements and the number of entries in the matrix is denoted by “SR” (sampling ratio). The ratio between the dimension of the set of $n \times n$ rank r matrices, $r(2n - r)$, and the number of samples p, is denoted by “FR”. In our tests, we fixed FR to 0.2 and 0.3 and r to 10. We tested five matrices with dimension $n = 100, 200, 300, 400, 500$ and set the number p to $r(2n - r)/FR$. The random seed for generating random matrices in MATLAB was set to 0. μ was set to $5n$ (a heuristic argument for this choice can be found in [5]). We set the step length τ to $1/\mu$ since for matrix completion problems $\|\mathcal{P}_\Omega\| = 1$. We terminated the code when the relative error between the residual and the true matrix was less than 10^{-4}, i.e.,

$$\|\mathcal{P}_\Omega(X^k) - \mathcal{P}_\Omega(M)\|_F/\|\mathcal{P}_\Omega(M)\|_F < 10^{-4}. \tag{5.6}$$

Note that this stopping criterion was used in [5]. We also set the maximum number of iteration to 2000.

We report the number of iterations needed by LB and ALB to reach (5.6) in Table 5.2. Note that performing the shrinkage operation, i.e., computing an SVD, dominates the computational cost in each iteration of LB and ALB. Thus, the per-iteration complexities of LB and ALB are almost the same and it is reasonable to compare the number of iterations needed to reach the stopping criterion. We report the relative error $err := \|X^k - M\|_F/\|M\|_F$ between the recovered matrix X^k and the true matrix M in Table

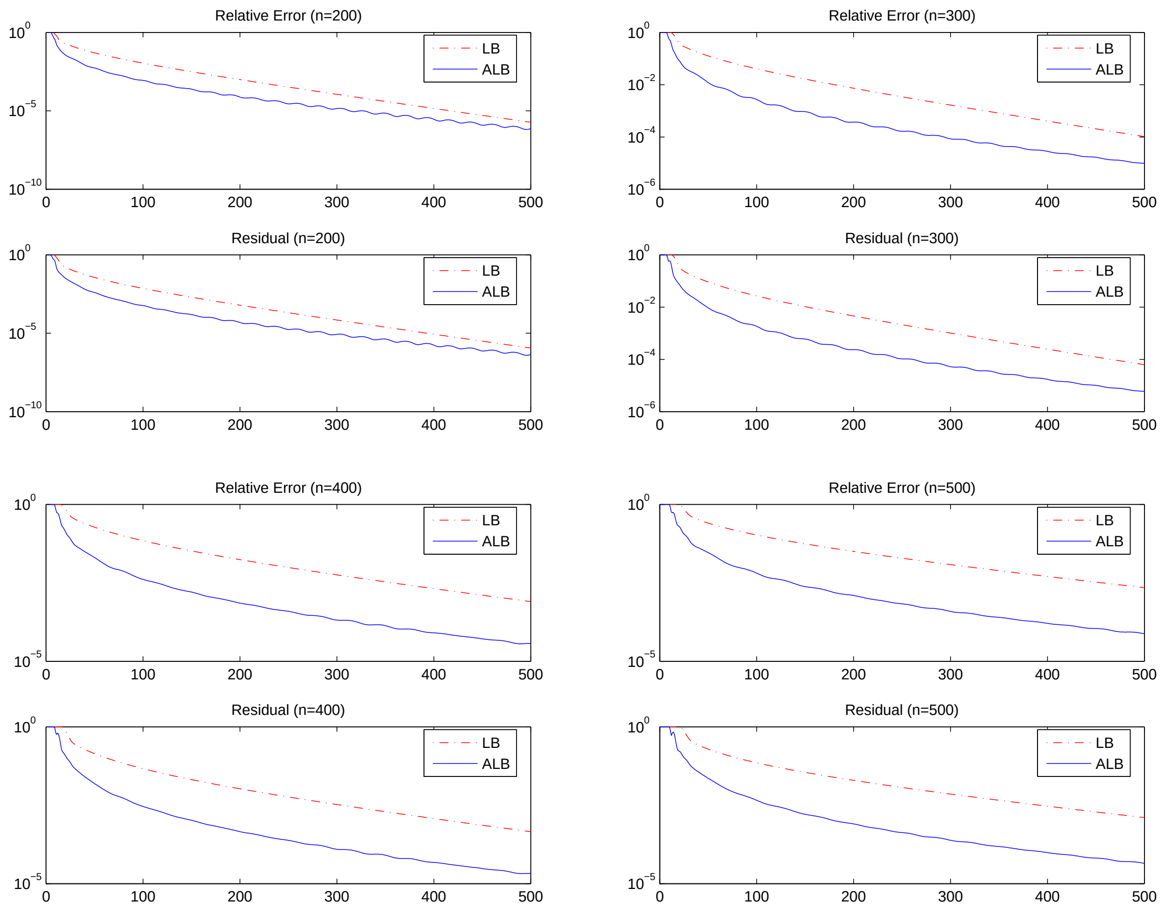

Fig. 5.4. *Comparison of LB and ALB on matrix completion problems with* rank $= 10, FR = 0.2$

5.2. We see from Table 5.2 that ALB needed significantly fewer iterations to meet the stopping criterion (5.6).

In Figures 5.4 and 5.5, we plot the Frobenius norms of the residuals and the relative errors obtained by LB and ALB for iteration 1-500 for the tests involving matrices with dimension $n = 200, 300, 400$ and 500. Note that the non-monotonicity of ALB is far less pronounced on these problems.

Table 5.2
Comparison between LB and ALB on Matrix Completion Problems

	$FR = 0.2, \text{rank} = 10$					$FR = 0.3, \text{rank} = 10$				
n	SR	iter-LB	err-LB	iter-ALB	err-ALB	SR	iter-LB	err-LB	iter-ALB	err-ALB
100	0.95	85	1.07e-4	63	1.11e-4	0.63	294	1.75e-4	163	1.65e-4
200	0.49	283	1.62e-4	171	1.58e-4	0.33	1224	3.76e-4	289	1.83e-4
300	0.33	466	1.64e-4	261	1.60e-4	0.22	2000+	3.59e-3	406	1.93e-4
400	0.25	667	1.79e-4	324	1.65e-4	0.17	2000+	1.12e-2	455	1.80e-4
500	0.20	831	1.76e-4	398	1.65e-4	0.13	2000+	3.14e-2	1016	7.49e-3

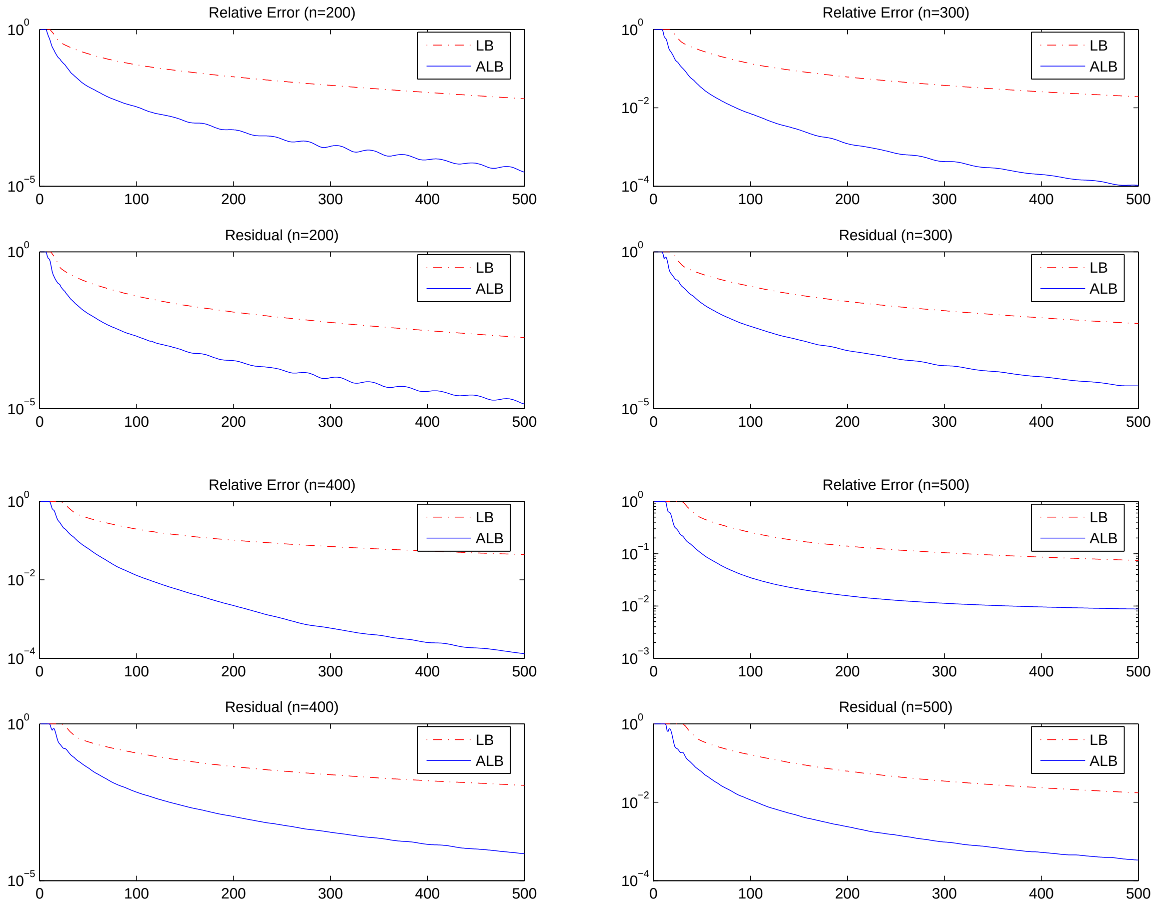

FIG. 5.5. *Comparison of LB and ALB on matrix completion problems with* $\text{rank} = 10, FR = 0.3$

6. Conclusions.

In this paper, we analyzed for the first time the iteration complexity of the linearized Bregman method. Specifically, we show that for a suitably chosen step length, the method achieves a value of the Lagrangian of a quadratically regularized version of the basis pursuit problem that is within ϵ of the optimal value in $O(1/\epsilon)$ iterations. We also derive an accelerated version of the linearized Bregman method whose iteration complexity is reduced to $O(1/\sqrt{\epsilon})$ and present numerical results on basis pursuit and matrix completion problems that illustrate this speed-up.

REFERENCES

[1] J. Barzilai and J. Borwein, *Two point step size gradient methods*, IMA Journal of Numerical Analysis, 8 (1988), pp. 141–148.

[2] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sciences, 2 (2009), pp. 183–202.

[3] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[4] L. Bregman, *The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming*, USSR Computational Mathematics and Mathematical Physics, 7 (1967), pp. 200–217.

[5] J. Cai, E. J. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM J. on Opti-

mization, 20 (2010), pp. 1956–1982.

[6] J.-F. Cai, S. Osher, and Z. Shen, *Convergence of the linearized Bregman iteration for ℓ_1-norm minimization*, Mathematics of Computation, 78 (2009), pp. 2127–2136.

[7] ———, *Linearized Bregman iterations for compressed sensing*, Mathematics of Computation, 78 (2009), pp. 1515–1536.

[8] E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Foundations of Computational Mathematics, 9 (2009), pp. 717–772.

[9] E. J. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Transactions on Information Theory, 52 (2006), pp. 489–509.

[10] E. J. Candès and T. Tao, *The power of convex relaxation: near-optimal matrix completion*, IEEE Trans. Inform. Theory, 56 (2009), pp. 2053–2080.

[11] D. Donoho, *Compressed sensing*, IEEE Transactions on Information Theory, 52 (2006), pp. 1289–1306.

[12] D. Goldfarb and S. Ma, *Fast multiple splitting algorithms for convex optimization*, tech. report, Department of IEOR, Columbia University. Preprint available at http://arxiv.org/abs/0912.4570, 2009.

[13] D. Goldfarb, S. Ma, and K. Scheinberg, *Fast alternating linearization methods for minimizing the sum of two convex functions*, tech. report, Department of IEOR, Columbia University. Preprint available at http://arxiv.org/abs/0912.4571, 2010.

[14] D. Goldfarb and K. Scheinberg, *Fast first-order methods for composite convex optimization with line search*, preprint, (2011).

[15] D. Gross, *Recovering low-rank matrices from few coefficients in any basis*, IEEE Transactions on Information Theory, 57 (2011), pp. 1548–1566.

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

[17] M. R. Hestenes, *Multiplier and gradient methods*, Journal of Optimization Theory and Applications, 4 (1969), pp. 303–320.

[18] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Mathematical Programming, Series B, 45 (1989), pp. 503–528.

[19] Y. Liu, D. Sun, and K.-C. Toh, *An implementable proximal point algorithmic framework for nuclear norm minimization*, To appear in Mathematical Programming, (2009).

[20] S. Ma, D. Goldfarb, and L. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Mathematical Programming Series A, 128 (2011), pp. 321–353.

[21] B. K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM Journal on Computing, 24 (1995), pp. 227–234.

[22] 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 (2005), pp. 229–251.

[23] Y. E. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence $\mathcal{O}(1/k^2)$*, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547.

[24] ———, *Introductory lectures on convex optimization*, 87 (2004), pp. xviii+236. A basic course.

[25] ———, *Smooth minimization for non-smooth functions*, Math. Program. Ser. A, 103 (2005), pp. 127–152.

[26] ———, *Gradient methods for minimizing composite objective function*, CORE Discussion Paper 2007/76, (2007).

[27] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, SIAM Journal on Multiscale Modeling and Simulation, 4 (2005), pp. 460–489.

[28] S. Osher, Y. Mao, B. Dong, and W. Yin, *Fast linearized Bregman iteration for compressive sensing and sparse denoising*, Communications in Mathematical Sciences, 8 (2010), pp. 93–111.

[29] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, New York, 1972, pp. 283–298.

[30] B. Recht, *A simpler approach to matrix completion*, To appear in Journal of Machine Learning Research., (2009).

[31] B. Recht, M. Fazel, and P. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Review, 52 (2010), pp. 471–501.

[32] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, 1970.

[33] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.

[34] ACM SIGKDD and Netflix, *Proceedings of kdd cup and workshop*, Proceedings available online at http://www.cs.uic.edu/ liub/KDD-cup-2007/proceedings.html.

[35] N. Srebro, *Learning with Matrix Factorizations*, PhD thesis, Massachusetts Institute of Technology, 2004.

[36] N. Srebro and T. Jaakkola, *Weighted low-rank approximations*, in Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), 2003.

[37] K.-C. Toh and S. Yun, *An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems*, Pacific J. Optimization, 6 (2010), pp. 615–640.

[38] P. Tseng, *On accelerated proximal gradient methods for convex-concave optimization*, submitted to SIAM J. Optim., (2008).

[39] E. van den Berg and M. P. Friedlander, *Probing the Pareto frontier for basis pursuit solutions*, SIAM J. on Scientific Computing, 31 (2008), pp. 890–912.

[40] Z. Wen, W. Yin, and Y. Zhang, *Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm*, preprint, (2010).

[41] W. Yin, *Analysis and generalizations of the linearized Bregman method*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 856–877.

[42] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing*, SIAM Journal on Imaging Sciences, 1 (2008), pp. 143–168.