Easily Parallelizable and Distributable Class of Algorithms for Structured Sparsity, With Optimal Acceleration

Seyoon Ko, Donghyeon Yu, and Joong-Ho Won

Department of Statistics, Seoul National University, Seoul, South Korea; Department of Statistics, Inha University, Incheon, South Korea

ABSTRACT
Many statistical learning problems can be formulated as minimization of a sum of two convex functions, one typically a composition of nonsmooth and linear functions. Examples include regression under structured sparsity assumptions. Popular algorithms for solving such problems, for example, ADMM, often involve nontrivial optimization subproblems or smoothing approximation. We consider two classes of primal–dual algorithms that do not incur these difficulties, and unify them from a perspective of monotone operator theory. From this unification, we propose a continuum of preconditioned forward–backward operator splitting algorithms amenable to parallel and distributed computing. For the entire region of convergence of the whole continuum of algorithms, we establish its rates of convergence. For some known instances of this continuum, our analysis closes the gap in theory. We further exploit the unification to propose a continuum of accelerated algorithms. We show that the whole continuum attains the theoretically optimal rate of convergence. The scalability of the proposed algorithms, as well as their convergence behavior, is demonstrated up to 1.2 million variables with a distributed implementation. The code is available at https://github.com/kose-y/dist-primal-dual. Supplementary materials for this article are available online.

1. Introduction
Many statistical learning problems can be formulated as an optimization problem of the form

$$\min_{x \in \mathbb{R}^p} f(x) + h(Kx),$$

where $K \in \mathbb{R}^{1 \times p}$, and both $f$ and $h$ are closed, proper, and convex. In this article, we assume $f$ is differentiable and its gradient $\nabla f$ is Lipschitz continuous with modulus $L_f$; $h$ is not necessarily smooth. Under this setting, we show how to solve Equation (1) in a fashion that is easy to parallelize or distribute on modern high-performance computing environment such as workstations equipped with multiple graphics processing units (GPUs).

A pinnacle instance of Equation (1) is high-dimensional penalized regression with structured sparsity:

$$\min_{x \in \mathbb{R}^p} \sum_{i=1}^n l_i(a_i^T x, b_i) + H(Dx),$$

with direct identification $f(x) = \sum_{i=1}^n l_i(a_i^T x; b_i)$, $H(u) = h(u)$, and $K = D$, where the set $\{(a_i, b_i) : a_i \in \mathbb{R}^p, b_i \in \mathbb{R}, i = 1, \ldots, n\}$ constitutes a training sample, $l_i : \mathbb{R}^2 \to \mathbb{R}$ is the loss function that may depend on the sample index, $D \in \mathbb{R}^{1 \times p}$ is the structure-inducing matrix, and $H$ is the penalty function, which is typically nonsmooth. Loss functions with Lipschitz gradients arise in many important problems: in linear regression we have $f(x) = (1/2)\|Ax - b\|_2^2$ and the gradient $\nabla f(x) = A^T (Ax - b)$ is $\|A^T A\|_2$-Lipschitz, where $A = [a_1, \ldots, a_n]^T$ denotes the data matrix and $\|\cdot\|_2$ is the standard operator norm with respect to the vector $\ell_2$ norm; in logistic regression $f(x) = -\sum_{i=1}^n \left( b_i(a_i^T x) + \log(1 + e^{a_i^T x}) \right)$ has $(1/4)\|A^T A\|_2$-Lipschitz gradients. Choosing the $\ell_1$-penalty $H(z) = \|z\|_1$ for some $\lambda > 0$ yields the generalized lasso (Tibshirani and Taylor 2011), which includes the fused lasso (Tibshirani et al. 2005) as a special case. For the group lasso (Yuan and Lin 2006) with $G$ possibly overlapping groups, we can choose $H(y) = \lambda_1 \|y_{[1]}\|_q + \cdots + \lambda_G \|y_{[G]}\|_q$ for $y = (y_{[1]}^T, \ldots, y_{[G]}^T)^T$, where $[g] \subset \{1, 2, \ldots, p\}$ is a given set of group indexes and $y_{[g]} \in \mathbb{R}|g|$ for each $g = 1, 2, \ldots, G$; $\| \cdot \|_q$ denotes the $\ell_q$ norm with $q > 1$. Now set $D$ as a $((1 \times 1) + \cdots + ([G]) \times p)$ binary matrix with a single one (1) in each row; the 1 corresponds to the group membership. Then,

$$H(Dx) = \lambda_1 \|x_{[1]}\|_q + \cdots + \lambda_G \|x_{[G]}\|_q$$

as desired; $D$ has a column with more than a single nonzero entry if and only if there is an overlapping group. Judicious choices of $f$, $h$, and $K$ in Equation (1) allow more flexibility in solving Equation (2). In particular, nonsmooth loss functions, such as the hinge loss, can also be handled. More complex penalty functions such as the latent group lasso (Jacob, Obozinski, and Vert 2009) are also allowed in Equation (2) (See Appendix A for details). Therefore, ability to solve Equation (1) efficiently provides a versatile tool for many important statistical learning problems.

In spite of its importance, solving Equation (1) is challenging because the non-separability of the nonsmooth part hampers use of efficient methods. If $K = I$ and $h$ is separable, for example,
\( h(y) = \lambda \|y\|_1 \), then the proximal gradient method (Combettes and Wajs 2005) is arguably the method of choice, which provides a simple gradient-descent-like iteration

\[
x^{k+1} = \arg \min_x f(x) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2t} \|x - x^k\|_2^2 + h(x) = \text{prox}_{th}(x^k - t\nabla f(x^k))
\]

for \( 0 < t < 2/L_f \), where \( \text{prox}_{\phi}(z) := \arg \min_{z' \in \mathbb{R}^n} \phi(z') + \frac{1}{2} \|z' - z\|_2^2 \) is the proximity operator for a convex function \( \phi \). If \( h(y) = \lambda \|y\|_1 \), then \( \text{prox}_{th} \) is an element-wise soft-thresholding operator (Beck and Teboulle 2009). However, for general \( K \) and other choices of \( h \), for example, group lasso, proximal gradient involves evaluating \( \text{prox}_{th} \), which is nontrivial even for tractable cases (Friedman et al. 2007; Liu, Yuan, and Ye 2010; Xin et al. 2014; Yu et al. 2015). While approximating \( h \) by a smooth function has been considered (Nesterov 2005; Chen et al. 2012), this approach introduces an additional smoothing parameter that is difficult to choose in practice. The popular alternating directions method of multipliers (ADMM; see, for example, Boyd et al. 2010) can be applied to solve Equation (1) as well, which yields an iteration

\[
x^{k+1} = \arg \min_x f(x) + (t/2)\|Kx - x^k + (1/t)y^k\|_2^2, \quad (3a)
\]

\[
\tilde{x}^{k+1} = \text{prox}_{(1/t)h}(Kx^{k+1} + (1/t)y^k), \quad (3b)
\]

\[
y^{k+1} = y^k + t(Kx^{k+1} - \tilde{x}^{k+1}). \quad (3c)
\]

The \( x \)-update (3a) is an inner minimization subproblem and is potentially expensive to compute. For example, if \( f \) is a loss function for a generalized linear model, then the corresponding update involves solving a linear equation of the form \((A^TWA + tK^TK)x = r\), \( W \) diagonal, iteratively. While \( K \) is structured and known a priori, the data matrix \( A \) is hard structured. A similar problem arises in medical imaging reconstruction problems, such as undersampled multi-coil MRI reconstruction (Ramani and Fessler 2011) or sparse-view CT reconstruction (Sidky, Jørgensen, and Pan 2012) using the total variation penalty (Rudin, Osher, and Fatemi 1992; Goldstein and Osher 2009). In this case, the “measurement matrix” \( A \) is large and unstructured. Hence, avoiding inner minimization subproblem is crucial in both statistical learning and imaging problems where the problem dimensions are ever increasing. Primal-dual hybrid gradient method (PDHG; Zhu and Chan 2008; Esser, Zhang, and Chan 2010; Chambolle and Pock 2011; He and Yuan 2012; Chambolle and Pock 2016; Zhu 2017) and linearized alternating directions method (LADM; Lin, Liu, and Su 2011) add an additional regularization term to Equation (3a) in order to avoid the costly inner minimization subproblem. However, these methods often involve evaluating \( \text{prox}_{h} \), which may lead to another inner minimization subproblem in the presence of \( A \).

The goal of this article is to introduce to the statistical community a class of algorithms that requires neither smoothing nor quadratic minimization. This class of algorithms only involve evaluation of the gradient \( \nabla f(x) \), matrix–vector multiplications and simple proximity operators. Thus, it is simple to implement and attractive for parallel and distributed computation. We begin with introducing two known algorithms. One is due to Loris and Verhoeven (2011), later studied by Chen, Huang, and Zhang (2013), and Drori, Sabach, and Teboulle (2015)

\[
x^{k+1} = x^k - \tau \left( \nabla f(x^k) + K^T y^k \right),
\]

\[
y^{k+1} = (1 - \rho_k)y^k + \rho_k \text{prox}_{\sigma \|\cdot\|_1}(y^k + \sigma Kx^{k+1}),
\]

\[
\tilde{x}^{k+1} = 2x^{k+1} - x^k,
\]

and the other is due to Condat (2013) and Vă (2013)

\[
x^{k+1} = x^k - \tau \left( \nabla f(x^k) + K^T y^k \right),
\]

\[
\tilde{x}^{k+1} = 2x^{k+1} - x^k,
\]

\[
y^{k+1} = (1 - \rho_k)y^k + \rho_k \text{prox}_{\sigma \|\cdot\|_1}(y^k + \sigma Kx^{k+1}),
\]

(Algorithm LV)

\( h^*(\nu) = \sup_{u \in \mathbb{R}^n} \langle u, \nu \rangle - h(u) \) is the convex conjugate of \( h \). Choices of the sequence \( \{\rho_k\} \) and the step size parameters \((\sigma, \tau)\) for convergence of these algorithms are discussed in Section 2. As can be seen, the proximity operator employed by both algorithms depends only on \( h^* \) but not \( K \). Moreover, \( \text{prox}_{\sigma \|\cdot\|_1} \) can be evaluated by using Moreau’s decomposition \( \text{prox}_{\sigma \|\cdot\|_1}(y) = y - \sigma \text{prox}_{\|\cdot\|_1}(y/\sigma) \). Thus, they are simple to implement and attractive for parallel and distributed computation as long as either \( \text{prox}_{\sigma \|\cdot\|_1} \) or \( \text{prox}_{\sigma \|\cdot\|_1} \) is simple (“proximable”). Table 1 illustrates the proximity operators for popular choices of \( h \). Once the conditions for convergence is understood, the rate of convergence and acceleration of the algorithm are the next interest.

In this regard, the contributions of this paper are as follows. First, we connect Algorithms LV and CV from a perspective of monotone operator theory to show that they are essentially the same preconditioned forward-backward (FB) splitting algorithm (see, for example, Combettes and Wajs 2005) sharing a

| Name | \( h(y) \) | \( h^*(z) \) | \( \text{prox}_{h^*}(z) \) |
|------|----------------|----------------|-------------------|
| \( \ell_1 \)-norm | \( \lambda \|y\|_1 \) | \( \delta_{B_\infty}(z) = \{z : \|z\|_\infty \leq \lambda\} \) | \( \min \{\max(z, -\lambda), \lambda\} \) |
| \( \ell_2 \)-norm | \( \lambda \|y\|_2 \) | \( \delta_{B_2}(z) = \{z : \|z\|_2 \leq \lambda\} \) | \( \rho_{B_2}(z) \) |
| \( \ell_\infty \)-norm | \( \lambda \|y\|_\infty \) | \( \delta_{B_\infty}(z) = \{z : \|z\|_\infty \leq \lambda\} \) | \( \rho_{B_\infty}(z) \) |
| \( \ell_1 \)-norm | \( \sum_{i=1}^p \|y_i\|_q \) | \( \delta_{G_{\ell_1}}(z, G) = \{z : \|z_i\|_q \leq \lambda\} \) | \( \{r_{G_{\ell_1}}(z, G)\}_{z \in G} \) |
| nuclear norm | \( \lambda \sum_i \|y_i\|_\infty \) | \( \delta_{G_{\ell_1}}(z, G) = \{z : \|z_i\|_2 \leq \lambda\} \) | \( \min \{\Sigma_{i=1}^P z_i, \lambda \|y\|_1 \leq \lambda \} \) |
| hinge loss | \( \sum_{i=1}^P \max\{1 - y_i, 0\} \) | \( \delta_{H_{\ell_1}}(z) = \{z : \|z_i\|_1 \leq \lambda\} \) | \( \min \{z + 1, \max(z, 1)\} \) |

Notes: Function \( \delta_S \) denotes the indicator function for set \( S \) so that \( \delta_S(u) = 0 \) if \( u \in S \) and \( \delta_S(u) = +\infty \) otherwise; \( P_S \) denotes the projection onto set \( S \), which is unique if \( S \) is closed and convex; \( \sigma_M(z) \) denotes the \( M \)th largest singular value of matrix \( M \). All min, max operations are elementwise. In \( \ell_1 \)-norm, \( 1/q + 1/s = 1 \).
common pre-conditioner. Second, from this connection we propose a new, broader family of pre-conditioners that generates an entire continuum of FB algorithms. Third, by a unified analysis, we show that this continuum of algorithms enjoys common ergodic and non-ergodic rates of convergence over the entire region of convergence. Prior to our connection the rates of the above two algorithms have been available under much more stringent conditions than that for convergence; we close this gap. Fourth, we proceed further to accelerate the whole continuum of algorithms to achieve the theoretically optimal rate of convergence. Only an optimal acceleration of Algorithm CV has been known (Chen, Lan, and Ouyang 2014), and acceleration of LV has remained an open problem. Finally, we demonstrate the scalability of the studied algorithms by implementing them on a distributed computing environment in case that data do not fit in the memory of a single device.

Organization. In Section 2, we examine the relation between Algorithms LV and CV and unify them to propose a broader class of algorithms. The rates of convergence of this class of algorithms is also analyzed. In Section 3, we develop an accelerated variant of the new class of algorithms achieving the optimal rate. Its stochastic counterpart, also possessing the optimal rate, is discussed in Section 4. Section 5 demonstrates the convergence behavior and scalability of the new algorithms through their multi-GPU implementations. Discussion and conclusion follow thereafter in Section 6. All the proofs of our results can be found in the supplementary material.

Notation. That a symmetric matrix \( M \) is positive (semi) definite is denoted by \( M > 0 \) \( (M \geq 0) \); \( L > 0 \) \( (L \geq 0) \) refers to \( L - M > 0 \), etc. For \( M > 0 \), we define its associated inner product and norm by \( \langle x, x' \rangle_M = \langle Mx, x' \rangle \) and \( \| x \|_M = \sqrt{\langle x, x \rangle_M} \). For a symmetric matrix \( M \), denote the maximum and minimum eigenvalues \( \lambda_{\text{max}}(M) \) and \( \lambda_{\text{min}}(M) \), respectively. For a convex function \( f : X \to \mathbb{R} \cup \{+\infty\} \), we define the effective domain \( \text{dom} f = \{ x \in X : f(x) < +\infty \} \).

2. Unification

In this section we provide a unified treatment to Algorithms LV and CV from the perspective of monotone operator theory. For a brief summary of monotone operator theory, see Appendix C.

2.1. Relation Between Algorithms LV and CV

It can be shown that both Algorithms LV and CV are instances of preconditioned FB splitting. To be specific, note the first-order optimality condition for Equation (1) is given by

\[
0 = \nabla f(x^*) + K^T y^*,
\]

\[
y^* \in \partial h(Kx^*).
\]

where \( \partial h(y) = \{ g \in \mathbb{R}^l : h(y + g) \geq h(y), \forall g \in \mathbb{R}^l \} \) is the subdifferential of the convex function \( h \) at \( y \), which is a set-valued operator. Since \( h \) is closed and proper, condition (4b) is equivalent to \( Kx^* \in (\partial h)^{-1}(y^*) = \partial h^*(y^*) \) (Bertsekas 2009), thus Equation (4) can be equivalently written as an inclusion problem

\[
\begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \begin{bmatrix} \nabla f & K^T \\ -K & \partial h^* \end{bmatrix} \begin{bmatrix} x^* \\ y^* \end{bmatrix} =: (T(z^*), z^* = (x^*, y^*). \tag{5}
\]

Under a mild condition (Bauschke and Combettes 2011, Theorem 19.1 and Proposition 19.18 see also Condat 2013), Equation (5) has a solution. If \( (x^*, y^*) \) is a solution, then it is a saddle point for the saddle point formulation of Equation (1):

\[
\min_{x \in \mathbb{R}^p} \max_{y \in \mathbb{R}^l} \mathcal{L}(x, y), \tag{6}
\]

where \( \mathcal{L}(x, y) = f(x) + (Kx, y) - h^*(y) \) is the saddle function. Also the strong duality holds: \( x^* \) is a primal solution to Equation (1), and \( y^* \) is a solution to the associated dual

\[
\max_{y \in \mathbb{R}^l} (-f^*(-K^T y) - h^*(y)).
\]

In the sequel, we assume that Problem (5) has a solution.

The set-valued operator \( T \) is split into \( T = F + G \), where

\[
F = \begin{bmatrix} 0 & K^T \\ -K & \partial h^* \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix}. \tag{7}
\]

The operator \( F \) is maximally monotone and \( G \) is 1/\( L_f \)-cocoercive (Bauschke and Combettes 2011). A preconditioned FB splitting for solving Problem (5) is

\[
z^{k+1} = (I + M^{-1}F)^{-1}(I + M^{-1}G)(z^k)
\]

\[
z^{k+1} = (1 - \rho_k)z^k + \rho_kz^k,
\]

for \( z^k = (x^k, y^k), \tilde{z}^k = (\tilde{x}^k, \tilde{y}^k), \) and \( M > 0 \). If the modulus of cocoercivity of \( M^{-1}G \) denoted by \( \gamma \) (cocoercivity of \( G \) is preserved; see Davis 2015), then Equation (8) converges if \( \gamma > 1/2 \) and for a sequence \( \{ \rho_k \} \subset [0, 1) \) such that \( \sum_{k=0}^{\infty} \rho_k(\delta - \rho_k) = \infty \) with \( \delta = 2 - 1/(2\gamma) \). Note \( \rho_k \equiv 1 \) is allowed which yields a simple iteration \( z^{k+1} = (I + M^{-1}F)^{-1}(I + M^{-1}G)z^k \).

The inverse operator \( (I + M^{-1}F)^{-1} \) is single-valued due to maximal monotonicity of \( M^{-1}F \) (Bauschke and Combettes 2011, Theorems 25.8 and 24.5). For instance, \( (I + \partial \phi)^{-1}(z) = \arg \min_{z' \in \mathbb{R}^p} \phi(z') + \frac{1}{2}\|z' - z\|^2 = \text{prox}_\phi(z) \). In particular, the preconditioners for Algorithms LV and CV are respectively given by Combettes et al. (2014), Condat (2013), and Vu (2013):

\[
M = M_{\text{LV}} := \begin{bmatrix} 1/\sigma & I - \tau KK^T \\ 1/\tau K & I \end{bmatrix}
\]

\[
M = M_{\text{CV}} := \begin{bmatrix} 1/\tau K & I - \tau KK^T \\ -K & I \end{bmatrix}.
\]

Now we are ready to see that Algorithms LV and CV are essentially the same algorithm. The "LDL" decomposition of \( M_{\text{CV}} \) reveals that

\[
M_{\text{CV}} = \begin{bmatrix} I & 1/\tau K \\ -\tau KK^T & I \end{bmatrix} \begin{bmatrix} 1/\sigma I & \tau KK^T \\ 1/\tau K & I \end{bmatrix} = M_{\text{LV}}L^T.
\]

It is clear both \( M_{\text{LV}} \) and \( M_{\text{CV}} \) are positive definite if and only if \( 1/(\tau \sigma) > \|K\|^2_2 \). Also, it is easy to see that Algorithm CV, that is, Equation (8) with \( M = M_{\text{CV}} \), is equivalent to

\[
L^T z^{k+1} = (1 - \rho_k)LT z^k + \rho_k(I + M_{\text{CV}}^{-1}G)^{-1} \times (I - M_{\text{LV}}G)(LT z^k),
\]

\[
T(z^*), \quad z^* = (x^*, y^*).
\]
where $\bar{F} = L^{-1}FL^{-T}$ and $\bar{G} = L^{-1}GL^{-T}$. Letting $w = L^{2}z$, we see that Algorithm CV is in fact Algorithm LV applied to the linearly transformed variable $w$ by splitting the similarly transformed operator $L^{-1}TL^{-T}$ into $\bar{F}$ and $\bar{G}$. The cocoercivity constant of $M_{\bar{G}}^{-1}$ is found by the following proposition.

**Proposition 1.** $M_{\bar{G}}^{-1}$ is $(1/\tau - \sigma \|K\|_{2}^{2})/L_{f}$-cocoercive with respect to $\| \cdot \|_{M_{\bar{G}}}$.

Thus, from the discussion below Equation (8), we have $\gamma = (1/\tau - \sigma \|K\|_{2}^{2})/L_{f}$ and $\delta = 2 - \frac{L_{f}}{\tau} \cdot \frac{1}{1/\tau - \sigma \|K\|_{2}^{2}}$. Then Algorithm CV converges if

$$
\frac{1}{\tau} > \frac{L_{f}}{2} \quad \text{and} \quad \left( \frac{1}{\tau} - \frac{L_{f}}{2} \right) \frac{1}{\sigma} > \|K\|_{2}^{2}.
$$

(11)

With respect to the untransformed sequence $\{z_{k}\}$, observe that $M_{\bar{G}}G$ is also $(1/\tau - \sigma \|K\|_{2}^{2})/L_{f}$-cocoercive (with respect to $\| \cdot \|_{M_{G}}$). In light of Equation (10), it is natural to measure convergence using the metric $\|L^{T} \cdot \|_{M_{G}}$, and this metric coincides with $\| \cdot \|_{\bar{M}}$. On the other hand, it is easy to see $M_{\bar{G}}G$ is $1/\tau L_{f}$-cocoercive with respect to $\| \cdot \|_{M_{\bar{G}}}$. Hence Algorithm LV has $\gamma = 1/(\tau L_{f})$ and $\delta = 2 - \tau L_{f}/2$. It converges if

$$
\frac{1}{\tau} > \frac{L_{f}}{2} \quad \text{and} \quad 1/(\tau \sigma) > \|K\|_{2}^{2}.
$$

(12)

Both Equations (11) and (12) recover the known convergence regions in the literature (Condat 2013; Chen, Huang, and Zhang 2013).

### 2.2. Unified Algorithm Class

The relation between the two algorithms suggests a more general family of preconditioners, namely

$$
M = \bar{L}M_{\bar{G}}\bar{L}^{T} = \begin{bmatrix} \frac{1}{\tau}I & 0 \\ C & \frac{1}{\sigma}I + \tau (CC^{T} - KK^{T}) \end{bmatrix},
$$

(13)

where $\bar{L}$ replaces $(2,1)$ block of $L$ in Equation (9) by $\tau C$. In particular, if $CK^{T} = KC^{T}$, then Equation (8) yields the following iteration (for simplicity we set $\rho_{k} \equiv 1$):

$$
y^{k+1} = \text{prox}_{\sigma h_{\gamma}}(\sigma Kx^{k} + \sigma (C - K)\nabla f(x^{k}) + (I + \sigma \tau (C - K)^{T})y^{k})
$$

$$
x^{k+1} = x^{k} - \tau (\nabla f(x^{k}) - C^{T}y^{k} + (C + K)^{T}y^{k+1}).
$$

(14)

Condition $CK^{T} = KC^{T}$ is satisfied if and only if $C = US^{-1}VT + NV^{T}$, where $U$, $V$, and $\Sigma$ are from the reduced singular value decomposition of $K = U\Sigma V^{T}$ so that $\Sigma$ is an $\tau \times \tau$ positive diagonal matrix, where $\tau \equiv \text{rank}(K)$; $\bar{V}$ is such that $\bar{V} = [V, \bar{V}]$ is orthogonal; $S$ is symmetric, and $N$ is arbitrary. A simple choice is $S = \kappa \Sigma^{2}$ for some $\kappa \in \mathbb{R}$ and $N = 0$, yielding $C = \kappa K$. Choosing $\kappa = 0$ and $-1$, respectively, recovers Algorithms LV and CV; for $\kappa = 1$, we have

$$
y^{k+1} = \text{prox}_{\sigma h_{\gamma}}(\sigma Kx^{k} + y^{k})
$$

$$
x^{k+1} = x^{k} - \tau \nabla f(x^{k}) - \tau K^{T}(2y^{k+1} - y^{k}),
$$

(16)

which is the dual version of Algorithm CV (Condat 2013, Algorithm 3.2). Another choice is to set $S = \pm \Sigma^{2}$ and $N$ so that $NN^{T}$ is diagonal. In this case $CC^{T} - KK^{T}$ reduces to a diagonal matrix, $C = [\bar{K}, N\bar{V}]$ where $\bar{K}$ is the first $r$ columns of $KV$. If the eigenspace of $K^{T}K$ is well known and multiplication with $\bar{V}$ can be computed fast, for example, the discrete cosine transform matrix for the fused lasso on a regular grid (Lee et al. 2017), this choice can be useful.

### 2.3. Convergence Analysis

#### 2.3.1. Region of Convergence

A condition for (8) with general $M$ to converge is

$$
M > \begin{bmatrix} \frac{L_{f}}{\tau} & 0 \\ 0 & 0 \end{bmatrix}
$$

(15)

which follows from Theorem 2 and Proposition 3 later in this section. Thus, with $M$ in Equation (13), the following region of convergence is obtained.

**Proposition 2.** Algorithm (14) converges for $(\sigma, \tau)$ such that

$$
\frac{1}{\tau} > \frac{L_{f}}{2} \quad \text{and} \quad \left( \frac{1}{\tau} - \frac{L_{f}}{2} \right) \left( \frac{1}{\sigma} - \frac{\tau \|K\|_{2}^{2}}{\sigma} \right) > \frac{\tau L_{f}}{2} \|C\|_{2}^{2}.
$$

(16)

Note that Equation (16) reduces to Equation (12) for Algorithm LV and to (11) for CV. In general for $C = \kappa K$, $\kappa \in [-1, 1]$, the region of convergence shrinks gradually from $|\kappa| = 0$ (LV) to $1$ (CV); see Figure 1. This extends the observation made in Section 2.1 regarding convergence conditions (12) and (11) to a continuum of algorithms between LV and CV.

**Remark 1.** Condat (2013) also considers an extension of Equation (1), which minimizes the three-function sum $f(x) + g(x) + h(Kx)$, with $g$ convex closed proper (not necessarily smooth). In this case, the second term of the first line of Algorithm CV is replaced by $\text{prox}_{\sigma_{\gamma}}(x^{k} - \tau (\nabla f(x^{k}) + K^{T}y^{k}))$. We call this extension Algorithm $CV_{\gamma}$. This algorithm is still a preconditioned FB splitting one with preconditioner $M_{\bar{G}}$, where the zero in the $(1,1)$ block of operator $F$ is replaced by $\partial g$, and converges under

Figure 1. Region of convergence in $(1/\sigma, 1/\tau)$. Boundaries correspond to $|\kappa| = 0$, 0.25, 0.5, 0.75, 1.
Equation (11). For this extended $F$, Equation (14) generates a feasible algorithm only when $C = \pm K$, that is, Algorithm CV+ or its dual. Nevertheless, for Algorithm LV, there is a three-function extension (Chen, Huang, and Zhang 2016).

2.3.2. Rates of Convergence

We now analyze the rates of convergence of the preconditioned FB splitting algorithm (8) for the preconditioner matrices $M$ of Equation (13). A pre-gap function $G(z, z) : = L(\tilde{x}, y) - L(x, y)$, where $z = (x, y)$ and $\tilde{z} = (\tilde{x}, \tilde{y})$, is used to measure the convergence of the objective value, because the duality gap $G^{\ast}(z) : = \sup_{x \in \mathbb{R}^p} G(z, x)$, $\mathbb{R} \subset \mathbb{R}^p \times \mathbb{R}^l$, guarantees that the pair $\tilde{z} = (\tilde{x}, \tilde{y})$ is a primal–dual solution to (6) if $G^\ast(z) \leq 0$. The rate of convergence of a gap function is typically analyzed in terms of an averaged solution sequence $\tilde{z}^N = \sum_{k=0}^N \alpha_k \tilde{x}^k / \sum_{k=0}^N \alpha_k$ for some positive sequence $\{\alpha_k\}$, yielding an ergodic rate. Ergodic rates are widely studied in the literature (Loris and Verhoeven 2011; Chen, Huang, and Zhang 2013; Bo and Csetnek 2015; Chambolle and Pock 2011, 2016), partly due to ease of analysis. Sometimes the unaveraged (last) solution sequence $\tilde{z}_k$ or $\{\tilde{z}_k\}$ is preferred as it tends to preserve the desired structural properties better than the ergodic counterpart. Analysis based on the unaveraged sequence yields the nonergodic rate (Davis 2015).

First, we establish an $O(1/N)$ ergodic convergence rate of the pre-gap evaluated for an average of the first $N$ terms of the sequence $\{(\tilde{x}^k, \tilde{y}^k)\}$:

**Theorem 1.** In iteration (8), let $\mu$ be a constant such that $\|x(0)\|^2_{M^{-1}} \leq (1/\mu)\|x\|^2_2$, for all $x \in \mathbb{R}^p$. Let $\alpha = (2\mu)/(4\mu - L_f)$ and denote $\tilde{z}^k = (\tilde{x}^k, \tilde{y}^k)$, $\tilde{z}^k = (\tilde{x}^k, \tilde{y}^k)$. Define $\tilde{z}^N = (\tilde{x}^N, \tilde{y}^N)$ with $\tilde{z}^N = \sum_{k=0}^N \rho_k \tilde{x}^k / \sum_{k=0}^N \rho_k$ and $\tilde{z}^N = \sum_{k=0}^N \rho_k \tilde{y}^k / \sum_{k=0}^N \rho_k$. Also let $\tilde{\rho} = \sup_{k \geq 0} \rho_k$. If $\mu > L_f/2$ and $\{\rho_k\}$ is chosen so that $0 < \rho_k < 1/\alpha$ for all $k$, then the following holds for all $z = (x, y) \in \mathbb{R}^p \times \mathbb{R}^l$:

$$G(\tilde{z}^N, z) \leq \frac{3}{2} \sum_{k=0}^N \left( \|z^0 - z\|^2_M + \frac{\alpha L_f}{(1 - \rho_k) \min(M)} \|z^0 - z^\ast\|^2_M \right),$$

where $z^\ast = (x^\ast, y^\ast)$ is a solution to Equation (6).

The key observation in proving Theorem 1 is the following lemma, also used in the proof of Theorem 2.

**Lemma 1.** For $\rho \in (0, 2)$, consider a relation $z^+ = (I + M^{-1}F)^{-1}(I - M^{-1}G)z^-$, $z_0 = (1 - \rho)z^- + \rho z^+$. Write $z_0 = (x_0, y_0)$, $z^+ = (x^+, y^+)$, $z^- = (x^-, y^-)$, all in $\mathbb{R}^p \times \mathbb{R}^l$. Then,

$$2\rho \ G(z^+, z) \leq \|z^+ - z\|^2_M - \|z_0 - z\|^2_M + (1 - 2/\rho)\|z^+ - z_0\|^2_M - \|z_0\|^2_M + (L_f/\rho)\|x^- - x_0\|^2_2, \quad \forall z = (x, y).$$

Now let $F(x) = f(x) + h(Kx)$ be the primal objective function and $F^\ast$ be the primal optimal value. For an important class of penalty functions $h$ including those for the generalized and group lasso, the following rate for primal suboptimality holds.

**Corollary 1.** Assume the conditions for Theorem 1. If $\text{dom}(h) = \mathbb{R}^l$, that is, $h$ does not take the value $+\infty$, then there exists a constant $C_1$ independent of $N$ such that for all $N$,

$$0 \leq F(\tilde{x}^N) - F^\ast \leq C_1 / \sum_{k=0}^N \rho_k.$$

Thus if $\{\rho_k\}$ is chosen so that $\inf_{k \geq 0} \rho_k > 0$, we obtain $O(1/N)$ convergence of the primal suboptimality.

The following theorem establishes the nonergodic counterpart of Theorem 1.

**Theorem 2.** For some $\nu > L_f/2$ and $\epsilon > 0$, suppose $M$ in iteration (8) satisfies

$$M \geq \begin{bmatrix} vI & \epsilon I \end{bmatrix}.$$ 

Let $\alpha = 2\nu/(4\nu - L_f)$ and $z^k = (x^k, y^k)$, $\tilde{z}_k = (\tilde{x}_k, \tilde{y}_k)$. If $\{\rho_k\}$ is chosen so that $0 < \rho_k < 1/\alpha$ for all $k$ and $\tau = \inf_{k \geq 0} \rho_k (1 - \alpha \rho_k) > 0$, then the following holds:

$$G(\tilde{z}_k, z) \leq \|z^0 - z\|^2_M + \|z^0 - z^{\ast}\|^2_M + \|z^0 - z\|^2_M / (\sqrt{k} + 1), \quad \forall z = (x, y) \in \mathbb{R}^p \times \mathbb{R}^l,$$

and additionally, $G(\tilde{z}_k, z) = o(1/\sqrt{k} + 1)$. Furthermore, if $\text{dom}(h) = \mathbb{R}^l$, then there exists a constant $C_2$ independent of $k$ such that $0 \leq F(\tilde{x}_k) - F^\ast \leq C_2 / \sqrt{k + 1}$ for all $k$ and $F(\tilde{x}_k) - F^\ast = o(1/\sqrt{k + 1})$.

**Remark 2.** The little-o result suggests that the non-asymptotic upper bound of the gap function may be conservative and the gap may diminish faster than the $1/\sqrt{k + 1}$ rate. The outcomes of the numerical experiments in Section 5 also suggest that the bound is not tight.

2.3.3. Closing the Gap

Here, we describe how our results close the gap in the literature between the conditions for convergence and those for the rate. The following fact helps understanding the conditions for Theorems 1 and 2:

**Proposition 3.** For $M > 0$ and a given $L_f > 0$, the following are equivalent.

1. For all $x \in \mathbb{R}^p$, there exists $\mu > L_f/2$ such that $\|x(0)\|^2_{M^{-1}} \leq (1/\mu)\|x\|^2_2$.
2. The condition (15) holds.
3. There exist $\nu > L_f/2$ and $\epsilon > 0$ such that $M \geq \begin{bmatrix} vI & \epsilon I \end{bmatrix}$.

That is, the conditions for Theorems 1 and 2 are both equivalent to Equation (15). This implies that the rates of convergence results in this section hold for $M$ in Equation (13) satisfying Equation (16). Thus, for the entire range of $(\alpha, \tau)$ for which Equation (14) converges, we have established an $O(1/N)$ ergodic and an $o(1/\sqrt{k + 1})$ nonergodic convergence rates for the objective values.

For Algorithm LV ($M = M_N$), Loris and Verhoeven (2011) obtain an $O(1/N)$ ergodic convergence rate for $f(x) = \frac{1}{2}\|Ax - b\|^2_2$. For general $f$, Chen, Huang, and Zhang (2013) show that Algorithm LV converges under Equation (12), but the rate is given only for strongly convex $f$ and full row rank $K$. This special case is not very interesting in statistical learning applications in
which \( f \) is almost always not strongly convex. To the best of our knowledge, our result for the rates of convergence for Algorithm LV and its variants (including the optimal accelerated one in the next subsection) without this impractical assumption is novel. For Algorithm CV \((M = M_0)\), our result extends the region of parameters for which ergodic converge rate is known from \((1/\tau - \sigma \|K\|^2)/L_f \geq 1\) (Chambolle and Pock 2016, Theorems 1 and 2) to the full range \((1/\tau - \sigma \|K\|^2)/L_f \geq 1/2\) of Equation (11). Therefore, we close the gap between the conditions for convergence and those for the rate.

Remark 3. An inspection of the proof of Lemma 1 asserts that the results of this section also holds for the extended \( F \) (see Remark 1). Thus, we close the gap for Algorithm CV+, the three-function extension, as well.

Remark 4. Davis (2015, Proposition 5.3) analyzes both ergodic and non-ergodic rates for general \( F \) and \( G \), under the condition \( M \geq \lambda I \) for some \( \lambda > 0 \). When applied to Equation (8), this analysis results in a convergence region smaller than that is allowed by Equation (15). Here, we exploit the special structure of \( G \) in Equation (7).

### 3. Optimal Acceleration

It is well known that first-order methods can be accelerated by introducing some “inertia” (Nesterov 2004; Beck and Teboulle 2009; Chen et al. 2012). For the saddle-point problem of the form of Equation (6), the optimal rate of convergence is known to be \( O(L_f/N^2 + \|K\|_2/N) \) in terms of the duality gap \( \gamma^* \), where \( N \) is the total number of iterations (Nesterov 2005; Chen, Lan, and Ouyang 2014). A natural question arises regarding whether the same optimal rate can be attained for the entire continuum of algorithms. In this section, we show that the answer is affirmative.

#### 3.1. Algorithms

Chen, Lan, and Ouyang (2014) devise an accelerated variant of Algorithm CV that achieves the theoretically optimal rate of convergence \( O(L_f/N^2 + \|K\|_2/N) \), where \( N \) is the total number of iterations

\[
\begin{align*}
\tilde{x}^k &= \tilde{x}^k + \theta_k(\tilde{x}^k - \tilde{x}^{k-1}), \\
x_m^k &= (1 - \rho_k)x^k + \rho_k\tilde{x}^k, \\
y^{k+1} &= \text{prox}_{\rho_k\theta_k}(\tilde{y} + \sigma_kK\tilde{x}^k), \\
\tilde{y}^{k+1} &= \tilde{x}^k - \theta_k(\nabla f(x_m^k) + K^T\tilde{y}^{k+1}), \\
x^{k+1} &= (1 - \rho_k)x^k + \rho_k\tilde{y}^{k+1}, \\
y^{k+1} &= (1 - \rho_k)y^k + \rho_k\tilde{y}^{k+1}.
\end{align*}
\]

Note an extrapolation step (17a) with a parameter \( \theta_k \), and a “middle” relaxation step (17b) are introduced. For Equation (14), we consider the following generalization:

\[
\begin{align*}
\tilde{u}^k &= K\tilde{x}^k - \theta_kA(\tilde{x}^k - \tilde{x}^{k-1}), \\
\tilde{v}^k &= K^T\tilde{y}^k + \theta_k(\tau_k^{-1}\tau_{k-1}(K + B)^T - B^T)(\tilde{y}^k - \tilde{y}^{k-1}), \quad (18a)
\end{align*}
\]

Step sizes \((\sigma_k, \tau_k)\) are allowed to depend on the iteration count \( k \). This algorithm reduces to Equation (14) (hence to Algorithms LV, CV, and in between) if \( A = -C, B = C, \rho_k \equiv 1, \theta_k \equiv 0, \sigma_k \equiv \gamma, \) and \( \tau_k \equiv \tau \), and to Chen, Lan, and Ouyang (2014) for \( A = -K \) and \( B = 0 \). The optimal rate of convergence of Algorithm (18) is established in Section 3.2. In particular, the optimal acceleration of Algorithm LV is new.

#### 3.2. Convergence Analysis

We first consider the case in which the bounds for \( \{x^k\}, \{y^k\} \) is known \( a \) priori. In this case, we can assume that the search space is \( Z = X \times Y \), where \( X \subseteq \mathbb{R}^p, Y \subseteq \mathbb{R}^q \) are both closed and bounded. Under this assumption, we have the following bound for the duality gap:

\[
\begin{align*}
\text{Theorem 3. Let } \{z^k\} = \{(x^k, y^k)\} \text{ be the sequence generated by Algorithm (18). Assume for some } \Omega_X, \Omega_Y > 0, \\
\sup_{x^k, y^k \in \mathbb{X}} \|x - x^k\|_2^2 \leq 2\Omega_X^2, \\
\sup_{y^k \in \mathbb{Y}} \|y - y^k\|_2^2 \leq 2\Omega_Y^2,
\end{align*}
\]

and the parameter sequences \( (\rho_k), (\theta_k), (\tau_k), \) and \( (\sigma_k) \) satisfy

\[
\begin{align*}
\rho_k^{-1} - 1 &= \rho_k^{-1}\theta_{k+1}, \\
\frac{1 - q}{\tau_k} - L_f\rho_k - \frac{1}{r}\|A\|_2^2\sigma_k \geq 0, \\
\frac{1 - q}{\sigma_k} - \tau_k\left(2\|K + A\|_2\|K + B\|_2 + \frac{1}{q}\|B\|_2^2\right) \geq 0
\end{align*}
\]

for some \( q \in (0, 1), r \in (0, 1) \). Further suppose that

\[
0 < \theta_k \leq \min(\tau_{k-1}/\tau_k, \sigma_{k-1}/\sigma_k), \max(\tau_{k-1}/\tau_k, \sigma_{k-1}/\sigma_k) \leq 1.
\]

Then for all \( k \geq 1 \),

\[
\gamma^*(z^{k+1}) \leq \frac{\rho_k}{\tau_k}\Omega_X^2 + \frac{\sigma_k}{\tau_k}\Omega_Y^2.
\]

For the following choice of the algorithm parameters, we obtain the claimed optimal convergence rate.

\[
\begin{align*}
\rho_k &= \frac{2}{k+1}, \\
\theta_k &= \frac{k+1}{k}, \\
\tau_k &= \frac{k}{\tau_{k+1} + \theta_k\tau_{k-1}(K + B)^T}, \\
\sigma_k &= \frac{\Omega_Y}{\|K\|_2\Omega_X},
\end{align*}
\]
\[ P = \frac{1}{1-q} \text{ and } Q = \max \left\{ \frac{a^2}{(1-q)^2}, \frac{2cd+b^2}{q} \right\}, \]
then
\[ G^*(z^*) \leq \frac{4PQ^2}{(1-q)^2} L_f + \frac{2Q\Omega}{k} \| K \|_2, \quad \forall k \geq 2. \quad (23) \]

**Remark 5.** For \( A = -K, B = 0 \), Equation (20) recovers the condition for Chen, Lan, and Ouyang (2014, Theorem 2.1) by putting \( r \to 1 \) and \( q \to 0 \). For \( A = -\kappa K = -B \), we obtain (1 - \( |\kappa| \))/\( \tau_k \geq L_f \rho_k + |\kappa|/\| K \|_2 \sigma_k / r \) and (1 - \( |\kappa| \))/\( \tau_k \geq 2 \kappa_1 \| K \|_2 \tau_k \) in particular for Algorithm LV (\( k = 0 \)), we have 1/\( \tau_k \) \( \geq L_f \rho_k \) and 1/(\( \tau_k \sigma_k \)) \( \geq 2 \| K \|_2 \) regardless of \( q \) and \( r \); this condition resembles Equation (12).

Now suppose the bounds for \( \{x_k^\ast\}, \{y_k^\ast\} \) are unavailable. In this case, the duality gap \( \sup_{z \in Z} G(z, z), Z = \mathbb{R}^p \times \mathbb{R}^q \), may be unbounded above. Instead, we define a perturbed gap function
\[ \tilde{G}(z, v) := \sup_{z \in Z} G(z, z) - \langle v, z - z \rangle. \quad (24) \]

There always exists a perturbation vector \( v \) such that Equation (24) is finite (Monteiro and Svaiter 2011). Thus we want to find a vanishing sequence of perturbation vectors \( \{v_k\} \) that makes \( \tilde{G}(z^k, v^k) \) small.

**Theorem 4.** Suppose that \( \{z^k\} = \{(x^k, y^k)\} \) are generated by Algorithm (18). If the parameter sequences \( \{\rho_k\}, \{\theta_k\}, \{\tau_k\}, \) and \( \{\sigma_k\} \) satisfy Equation (20) and
\[ \theta_k = \tau_{k-1}/\tau_k = \sigma_{k-1}/\sigma_k \leq 1 \quad (25) \]
for some \( 0 < q < 1, 0 < r < 1/2 \). Then there exists a vector \( v^{k+1} \) such that for any \( k \geq 1 \),
\[ \tilde{G}(z^{k+1}, v^{k+1}) \leq \frac{\theta_k}{\tau_k} \left( 2 + \frac{q}{1-q} + \frac{r+1/2}{1-2r} \right) R^2 \]
\[ =: \epsilon_{k+1}, \text{ and} \quad (26) \]
\[ \|v^{k+1}\|_2 \leq \frac{\alpha_k}{\tau_k} \left( \|\hat{x} - \hat{x}^1\|_2 + \|\hat{y} - \hat{y}^1\|_2 \right) \]
\[ + \left( \frac{\alpha_k}{\tau_k} (\mu + \frac{q}{1-q}) + 2\rho_k (\mu \|A\|_2 + \nu \|B\|_2) \right) \]
\[ + 2\tau_k \rho_k v \|K + A\|_2 |K + B\|_2 R, \quad (27) \]
where \( (\hat{x}, \hat{y}) \) is a pair of solutions to problem (6), and
\[ R = \sqrt{\|\hat{x} - \hat{x}^1\|^2 + \frac{\mu}{\alpha_k} \|\hat{y} - \hat{y}^1\|^2}, \quad \mu = \sqrt{\frac{1}{1-q}}, \quad (28) \]

For the following choice of the algorithm parameters, we obtain the claimed optimal convergence rate.

**Corollary 3.** If \( \|A\|_2 \leq a \|K\|_2, \|B\|_2 \leq b \|K\|_2, \|K + A\|_2 \leq c \|K\|_2, \) and \( \|K + B\|_2 \leq d \|K\|_2 \) for some \( a, b, c, d > 0 \), \( N \) is given, and the parameters are set to
\[ \rho_k = \frac{2}{k+1}, \quad \theta_k = \frac{k+1}{k}, \quad \tau_k = \frac{k}{2PQ + QN \|K\|_2^2}, \quad \sigma_k = \frac{k}{N \|K\|_2}, \quad (29) \]
where
\[ P = \frac{1}{1-q}, \quad Q = \max \left\{ \frac{a^2}{(1-q)^2}, \frac{2cd+b^2}{q} \right\}, \]
then
\[ \epsilon_{N+1} \leq \left( \frac{4PL_f}{N^2} + \frac{2Q \|K\|_2}{N} \right) \left[ 2 + \frac{q}{1-q} + \frac{r+1/2}{1-2r} \right] R^2, \quad (30) \]
\[ \|v^{N+1}\|_2 \leq \frac{4PQ^2}{N^2} \left( \|\hat{x} - \hat{x}^1\|_2 + \|\hat{y} - \hat{y}^1\|_2 \right) + R \left( \mu + \frac{\tau_1}{\sigma_1} \right) \]
\[ + \frac{\|K\|_2}{N} \left[ 2Q \left( \|\hat{x} - \hat{x}^1\|_2 + \|\hat{y} - \hat{y}^1\|_2 \right) + R \left( \mu + \frac{\tau_1}{\sigma_1} \right) \right] \]
\[ + 4R(\mu a + b v) + \frac{4R \alpha v}{Q}. \quad (31) \]

This result can be interpreted as follows. Theorem 4 and Corollary 3 state that for every pair of positive scalars \( (\rho, \epsilon) \), Algorithm (18) generates \( (x^N, y^N) \) such that \( \|v^N\| \leq \rho \) and \( \epsilon_N \leq \epsilon \) (see Equations (26), (27), (30), and (31)) for a sufficiently large \( N \). The associated pair \((x^N, y^N)\) is called a \((\rho, \epsilon)-\)saddle point of the unperturbed saddle point problem (6) (Monteiro and Svaiter 2011, Definition 3.10). With this notion, the following proposition can be stated.

**Proposition 4.** Under the assumptions of Theorem 4 and Corollary 3, there exists a vector \( w^N = (w^N_1, w^N_2) \) such that \( w^N \in T_{x^N}(x^N, y^N) \) and \( \|w^N\| \leq \rho + \sqrt{4L} \epsilon \) for some constant \( L > 0 \), where
\[ T_{x^N} = \left[ \nabla f \quad K^T \right] \frac{\partial_{\hat{x}} h^*}{\partial_{\hat{y}} h^*}. \]
Here, \( \partial_{\hat{x}} h^* \) is the \( \epsilon \)-subgradient of \( h^* \) defined as \( \partial_{\hat{x}} h^* (y) = \{ g : h^*(y) \geq h^*(y) + \langle y - y', g \rangle - \epsilon, \forall y' \in \mathbb{R}^q \}, \forall y \in \mathbb{R}^p \).

The condition \( w^N \in T_{x^N}(x^N, y^N) \) in Proposition 4 can be written as the following two inequalities:
\[ 0 \geq -\left( \nabla f (x^N) + K^T y^N, x - x^N \right) \]
\[ + \langle w^N_1, x - x^N \rangle - \epsilon_N, \forall x, \quad (32a) \]
\[ h^*(y) \geq h^*(y^N) + \langle Kx^N, y - y^N \rangle \]
\[ + \langle w^N_2, y - y^N \rangle - \epsilon_N, \forall y. \quad (32b) \]

Comparing with the optimality conditions (4) for the unperturbed saddle point problem (6):
\[ 0 \geq -\left( \nabla f (x^*) + K^T y^*, x - x^* \right) \]
\[ + \langle w^*, x - x^* \rangle - \epsilon_N, \forall x, \quad (32a) \]
\[ h^*(y) \geq h^*(y^*) + \langle Kx^*, y - y^* \rangle \]
\[ + \langle w^*, y - y^* \rangle - \epsilon_N, \forall y. \quad (32b) \]
we see that the sum of the last two terms in each of the right-hand sides of Equations (32a) and (32b) is the error of the approximate solution \((x^N, y^N)\). Indeed, in the unit ball centered at \((x^N, y^N)\), each error is bounded by \( \rho + \sqrt{4L} \epsilon + \epsilon \), which can be made arbitrarily small since the choice of \((\rho, \epsilon)\) is free. In this sense, for large \( N \), \((x^N, y^N)\) is a “nearly optimal” primal–dual solution.

### 4. Stochastic Optimal Acceleration

#### 4.1. Algorithm

In large-scale ("big data") applications, it is often the case that even the first-order information on the objective of Equation
(1) or (6) cannot be obtained exactly. Such settings can be modeled by a stochastic oracle, which provides unbiased estimators of the first-order information. To be precise, at the $k$th iteration suppose the oracle returns the stochastic gradient $(\tilde{\nabla}(x^k), \tilde{\nabla}_x(x^k), \tilde{\nabla}_y(y^k))$ independently from the previous iteration, such that
\[
\mathbb{E}[\tilde{\nabla}(x^k)] = \nabla f(x^k), \quad \mathbb{E}
\left[\begin{pmatrix}
\tilde{\nabla}_x(x^k) \\
\tilde{\nabla}_y(y^k)
\end{pmatrix}
\right] = K^T \hat{x}^k, \quad \mathbb{E}[\tilde{\nabla}_y(y^k)] = A^T \tilde{y}^k.
\]
We further assume that the variance of these estimators are uniformly bounded, that is,
\[
\mathbb{E}[\|\tilde{\nabla}(x^k)\|^2] \leq \chi_{2,F}, \quad \mathbb{E}[\|\tilde{\nabla}_x(x^k) - K^T \hat{x}^k\|^2] \leq \chi_{2,F}, \quad \mathbb{E}[\|\tilde{\nabla}_y(y^k)\|^2] \leq \chi_{2,F},
\]
and
\[
\mathbb{E}[\|\tilde{\nabla}_y(y^k) - A^T \tilde{y}^k\|^2] \leq \chi_{2,F}.
\]
(33)

For notational convenience, we define $\chi_x := \sqrt{\chi_{2,F}} + \chi_{2,K}$. We consider the following stochastic variant of (18):
\[
\bar{\mathbf{u}}_k = \tilde{K}_x \tilde{x}^k - \theta_k \tilde{K}_y \tilde{y}^k - \bar{\mathbf{u}}^k \tilde{\nabla}_x(x^k) + \beta \left( \frac{1}{\tau_k} \tilde{y}^k - \bar{\mathbf{v}}^k \right),
\]
\[
\bar{\mathbf{v}}_k = \tilde{K}_y \tilde{y}^k + \frac{\theta_k}{\tau_k} \bar{\mathbf{v}}^k + \tilde{\nabla}_y(y^k) + \beta \left( \frac{1}{\tau_k} \tilde{x}^k - \frac{1}{\tau_k} \bar{\mathbf{u}}^k \tilde{\nabla}_x(x^k) \right),
\]
\[
\tilde{x}^{k+1}_{md} = (1 - \rho_k) \tilde{x}^k + \rho_k \bar{\mathbf{u}}^k,
\]
\[
\bar{\mathbf{u}}^{k+1} = \tilde{\nabla}_x(x^k) + \bar{\mathbf{u}}^k,
\]
\[
\tilde{\mathbf{y}}^{k+1} = \text{prox}_{\sigma \tilde{\nabla}_y(y^k)} (\tilde{\mathbf{y}}^{k+1} - \theta_k \tilde{\mathbf{y}}^k),
\]
\[
\tilde{\mathbf{y}}^{k+1} = \text{prox}_{\sigma \tilde{\nabla}_y(y^k)} (\tilde{\mathbf{y}}^{k+1} - \theta_k \tilde{\mathbf{y}}^k).
\]
(34)

which can be considered a generalization of the stochastic variant of Equation (17) by Chen, Lan, and Ouyang (2014). The optimal rate of convergence of solving Equation (6) stochastically is known to be $O\left(\frac{L_j}{N^2} + \frac{\|K\|_2}{N} + \frac{x_t + x_y}{\sqrt{N}}\right)$ in terms of the expected duality gap $\mathbb{E}[\mathcal{G}(\mathcal{G}^*(z^i))]$ (Chen, Lan, and Ouyang 2014). In the sequel, we show that Algorithm (34) achieves this rate.

4.2. Convergence Analysis

We obtain the following results for Algorithm (34) when $Z$ is bounded. Note part (ii) of Theorem 5 is strengthened under the tail assumption:
\[
\mathbb{E}[\exp(\|\nabla f(x) - \tilde{\nabla}(x)\|^2/\chi_{2,F}^2)] \leq \exp(1),
\]
\[
\mathbb{E}[\exp(\|Kx - \tilde{\nabla}_x(x)\|^2/\chi_{2,F}^2)] \leq \exp(1),
\]
\[
\mathbb{E}[\exp(\|K^T y - \tilde{\nabla}_y(y)\|^2/\chi_{2,F}^2)] \leq \exp(1).
\]
Observe that Equation (35) implies Equation (33) by Jensen's inequality.

Theorem 5. Assume that Equation (19) holds, for some $\Omega_X, \Omega_Y > 0$. Also suppose that for all $k \geq 1$, the parameters $\rho_k, \theta_k, \tau_k$, and $\sigma_k$ in Equation (34) satisfy Equations (20a), (21),
\[
\begin{align}
&\frac{s-q}{\tau_k} - L_f \rho_k - \frac{\|A\|_2^2 \sigma_k}{r} \geq 0, \\
&\frac{t-r}{\sigma_k} - \tau_k \left( 2\|K\|_2^2 + 2\|B\|_2^2 + \frac{\|B\|_2^2}{q} \right) \geq 0
\end{align}
\]
for some $q, r, s, t \in (0, 1)$. Then the following holds.

(i) Under Assumption (33), we have $\mathbb{E}[\mathcal{G}(\mathcal{G}^*(z^{k+1}))] \leq C_0(k)$ for all $k \geq 1$, where
\[
C_0(k) := \frac{\rho_k}{\gamma_k} \left( \frac{2y_k}{\tau_k} \Omega_X^2 + \frac{2y_k}{\tau_k} \Omega_Y^2 \right) + \frac{\rho_k}{\gamma_k} \sum_{i=1}^k \left( \frac{(2-s)\gamma i}{1-s} \frac{1}{\gamma^2} \left( x^2_i + \frac{1}{\gamma^2} y^2_i \right) \right),
\]
\[
Q_1(k) := \frac{\rho_k}{\gamma_k} \left( \sqrt{2} \mathcal{X}_x \Omega_X + \mathcal{X}_y \Omega_Y \right)^2 \sum_{i=1}^k \gamma_i^2
\]
\[
+ \frac{\rho_k}{\gamma_k} \sum_{i=1}^k \left( \frac{(2-s)\gamma i}{1-s} \frac{1}{\gamma^2} \left( x^2_i + \frac{1}{\gamma^2} y^2_i \right) \right).
\]

Corollary 4. Assume condition (19) holds. In Algorithm (34), if $N \geq 1$ is given, $A = -K, |B|_2 \leq b|K|_2$, and the parameters are set to
\[
\rho_k = \frac{2}{k+1}, \quad \theta_k = \frac{k-1}{k}, \quad \tau_k = \frac{\Omega_k}{2L_f \Omega_X + Q|K|_2 \Omega_Y(N+1) + \mathcal{X}_x \sqrt{N-1}}, \quad \sigma_k = \frac{\Omega_k}{\|K\|_2 \Omega_X(N+1) + \mathcal{X}_y \sqrt{N-1}},
\]
where $P$ and $Q$ satisfies
\[
P = \frac{1}{\gamma^2} - Q \geq \max \left\{ \frac{1}{\gamma^2 - \gamma}, \frac{1}{\gamma^2} - \frac{1}{\gamma^2} \right\},
\]
the following holds.

(ii) Under assumption (35), then we have
\[
\mathcal{P}(\mathcal{G}^*(z^n)) \geq C_0(N) + \lambda C_1(N)
\]
\[
\leq 3 \exp(\lambda^2/3) + 3 \exp(-\lambda),
\]
for all $\lambda > 0$, where
\[
C_1(N) = \left( 1 + \frac{2-s}{3(1-s)} \right) \frac{\mathcal{X}_x \Omega_X}{\sqrt{N-1}} + \left( 1 + \frac{2-s}{3(1-s)} \right) \frac{\mathcal{X}_y \Omega_Y}{\sqrt{N-1}}.
\]
When $Z$ is unbounded, we have the following theorem.

**Theorem 6.** Assume that $\{z^k\} = \{(x^k, y^k)\}$ is the sequence generated by Equation (34). Further assume that the parameters $\beta_k$, $\theta_k$, $\tau_k$, and $\sigma_k$ in Equation (34) satisfy Equations (20a), (25), and (36), for all $k \geq 1$ and some $g, s, t \in (0, 1), r \in (0, 1/2)$. When Assumption (33) holds, there is a sequence of perturbation vectors $v^k$ satisfying

$$
\mathbb{E}[G(z^{k+1}, v^{k+1})] \leq \frac{\rho_k}{\tau_k} \left( 6 + \frac{4q}{1-q} + \frac{4(r+1/2)}{1/2-r} \right) R^2 + \left( \frac{2}{3} + \frac{2q}{1-q} + \frac{2(r+1/2)}{1/2-r} \right) S^2 \right]
$$

for all $k \geq 1$. Furthermore,

$$
\mathbb{E}[\|z^{k+1}\|_2^2, \|y^{k+1}\|_2^2] \leq \frac{2\rho_k}{\tau_k} \|\hat{x} - x^1\|_2^2 + \frac{2\rho_k}{\tau_k} \|\hat{y} - y^1\|_2^2 + \sqrt{2R^2 + S^2} \left( \frac{\rho_k (1 + \mu)}{\tau_k} + \left( \frac{1}{\sigma_1} + \frac{1}{\sigma_k} \right) \frac{\rho_k}{\sigma_k} \right) + 2\rho_k (\|A\|_2 + \|B\|_2) + 2\tau_k \rho_k \|K + A\|_2 K + B\|_2^2 \right]
$$

where $(\hat{x}, \hat{y})$ is a pair of solutions for Equation (6), $R, \mu$, and $\nu$ are as defined in Equation (28), and

$$
S := \sum_{i=1}^{k} \left( \frac{(2-t)_{\tau_i}(x_i^2 + x_{i+1}^2)}{1-t} \right) - \sum_{i=1}^{k} \left( 2-t_{\tau_i} = \frac{x_i^2 + x_{i+1}^2}{1-t} \right) = \frac{(2-t)_{\tau_i}(x_i^2 + x_{i+1}^2)}{1-t} - \sum_{i=1}^{k} \left( 2-t_{\tau_i} = \frac{x_i^2 + x_{i+1}^2}{1-t} \right)
$$

where $\tau = 2PLf + Q\|K\|_2(N - 1) + N\sqrt{N - 1}x/R$ for some $R > 0$, where $x$ is defined by $x = \sqrt{\frac{1}{2} - \frac{1}{t}} x^2 + \frac{1}{t - 1} x^2$. Then for $P$ and $Q$ satisfying

$$
P = \frac{1}{s^q}, \quad Q \geq \max \left\{ \frac{1}{s^{(r-t)q}}, \frac{b^2}{q(t-r)}, 1 \right\}
$$

we have

$$
\epsilon_N \leq \frac{4PLf}{N(N - 1)} + \frac{2Q\|K\|_2}{N} + \frac{2\sqrt{x/R}}{\sqrt{N - 1}}
$$

and

$$
\mathbb{E}[\|v^N\|_2^2] \leq \frac{4PLf}{N(N - 1)} + \frac{2Q\|K\|_2}{N} + \frac{2\sqrt{x/R}}{\sqrt{N - 1}}
$$

Further, we have $\epsilon_N \leq \min_{x} \frac{1}{2}\|b - Ax\|_2^2 + \lambda \|Dx\|_1$. Therefore, we obtain the desired order for both $\epsilon_N$ and $\mathbb{E}[\|v^N\|_2^2]$.

## 5. Numerical Experiments

### 5.1. Model Problems

**Overlapping group lasso.** We consider an overlapping group lasso problem with a quadratic loss

$$
\min_{x} \frac{1}{2}\|b - Ax\|_2^2 + \lambda \sum_{j=1}^{R} \sqrt{|g_j|\|x_{g_j}\|_2},
$$

where $A = [a_1, \ldots, a_n]^T$ is the data matrix, and $b = (b_1, \ldots, b_n)$ is the response vector. We generated a test dataset based on the methods in Chen et al. (2012). We defined $R$ groups of $S$ adjacent variables, with $10$ overlaps of adjacent groups. That is, $g_j = (90j - 10, \ldots, 90j + 10)$, thus $r = (R(S - 10) + 10)$. We set $x_j = (-1)^j \exp(- (j - 1)/100)$ for $j = 1, \ldots, p$. We sampled each element of $A$ from the standard normal distribution, and added Gaussian noise $\epsilon \sim N(0, 1)$ to $A$ to generate $b = Ax + \epsilon$. For the convergence experiments, we used $R = 100$ and $S = 100$, so that the dimension is given by $p = 9010$. For the scalability experiment, we used $S = 130$ and $R = 1000, 5000, 8000, 10000$ so that the dimensions are $p = 120,010, 600,010, 960,010, 1,200,010$. For all experiments, we set $n = 5000$ and $\lambda = R/100$.

**Graph-guided fused lasso.** The graph-guided fused lasso problem is considered by

$$
\min_{x} \frac{1}{2}\|b - Ax\|_2^2 + \lambda \|Dx\|_1,
$$

where $D$ is the difference matrix imposed by the network structure. The dataset for the graph-guided fused lasso experiments was generated following the transcription factor (TF) model of Zhu (2017). This is a simple gene network model with $J$ fully connected subnetworks of size $T$, where each subgroup has one TF with $T - 1$ regulatory target genes. Variables corresponding to TFs are sampled independently from $\mathcal{N}(0, 1)$. For $p$ variables, we sample each element of $A$ from the standard normal distribution, and added Gaussian noise $\epsilon \sim N(0, 1)$ to $A$ to generate $b = Ax + \epsilon$. For the convergence experiments, we used $R = 100$ and $S = 100$, so that the dimension is given by $p = 9010$. For the scalability experiment, we used $S = 130$ and $R = 1000, 5000, 8000, 10000$ so that the dimensions are $p = 120,010, 600,010, 960,010, 1,200,010$. For all experiments, we set $n = 5000$ and $\lambda = R/100$.
Figure 2. Convergence of the FB algorithms generated by Equation (14) and their accelerated variants (18) for a group lasso model (a)–(c) and a graph-guided fused lasso model (d)–(f). (a), (d), optimal acceleration with bounded parameter setting (“optimal”) with ergodic convergence of the FB algorithm (“base”). (b), (e), optimal acceleration with unbounded parameter setting (“optimal”) with ergodic convergence of the FB algorithm (“base”). (c), (f), nonergodic convergence of the FB (“base”) and inertial FBF (“inertial fbf”) algorithms. Solid black lines represent $O(1/k^2)$ convergence, and dashed black lines represent $O(1/k)$ convergence.

where $J_a$ is the number of active groups. Response $b_i$ is sampled so that $b_i = Ax + \epsilon_i$, with $\epsilon_i \sim \mathcal{N}(0, 100^2)$. In addition to the edges comprised of fully connected subnetworks, we added random edges between the active variables and the inactive variables. For each active variable, we added edges connecting this variable and $J - 1$ distinct inactive variables. For the convergence experiments, we used $T = 10$, $J_a = 20$, $J = 1000$ so that the dimension $p$ is 10,000. For the scalability experiment, we set $T = 12$, and $J_a = 20$. We selected $J = 10,000, 50,000, 80,000, 100,000$ to generate the dataset with $p = 120,000, 600,000, 960,000, 1,200,000$, respectively. For all experiments, we set $n = 5000$ and $\lambda = 1$.

5.2. Convergence Behavior

We applied the algorithms to the overlapping group lasso and graph-guided fused lasso with a quadratic loss, as described in Section 5.1. For the FB splitting (14), we set $C = \kappa K, |\kappa| \leq 1$. We set $\rho_k = 0.9 \left(2 - \frac{\tau L_1}{\alpha (1 - \kappa^2)} r_\sigma \|K\|_2^2 \right)$. Step sizes were chosen
as \( \tau = 0.99 \frac{1}{L_f} \) and \( \sigma = 0.9 \frac{1 \times L_f^2}{1 - (1-x^2) \sigma^2 L_f^2} \), so that Equation (16) is satisfied. For the acceleration (18), we tested four cases: Algorithm LV (\( A = B = 0 \)), CV (\( A = -K, B = K \)), their "midpoint" (\( A = -0.5K, B = 0.5K \)), and Chen, Lan, and Ouyang (2014) (\( A = -K, B = 0 \)). Number of iterations \( N \) is set to 10000. For bounded (Corollary 2) and unbounded (Corollary 3) cases, we found \((q, r)\) that minimizes
\[
\min_{q, r} A_{q, r} + 20 \frac{1}{Q} \|K\|_2 \text{ in Equation (23) and (} \frac{4PL_f}{N^2} + 20 \frac{1}{Q} \|K\|_2 \frac{1}{Q} \|K\|_2 \text{ in Equation (30)},
\]
respectively. Those minimizers were found using sequential least-squares programming. As a benchmark, we also applied an inertial version of the forward–backward–forward (FBF) algorithm (Combettes and Pesquet 2012) as described in Boţ and Csetnek (2016):

\[
\begin{align*}
\tilde{x}^{k+1} &= x^k - \tau \left( \nabla f(x^k) + K^T y^k \right) + \alpha_1 (x^k - x^{k-1}), \\
\tilde{y}^{k+1} &= \text{prox}_{\tau \rho} (y^k + \tau x^{k+1}) + \alpha_1 (x^k - y^{k-1}), \\
y^{k+1} &= y^k + \tau K \tilde{x}^{k+1} - x^{k+1} + \alpha_2 (y^k - y^{k-1}), \\
x^{k+1} &= x^k - \tau K^T (y^{k+1} - y^k) + \alpha_2 (x^k - x^{k-1}).
\end{align*}
\]

With \( \alpha_1 = \alpha_2 = 0 \), Algorithm (37) resembles Algorithm LV, but requires one more step per iteration; its convergence rate has not been established.

Figures 2(a), 2(b), 2(d), and 2(e) show the convergence of the FB (14) with respect to the averaged sequence \( \{\tilde{x}^n, \tilde{y}^n\} \), and the convergence of the accelerated FB algorithms (18) with respect to \( \{x^n, y^n\} \). We plot the gap between the primal objective value at \( x^k \) and the "optimal" objective value versus iteration count \( k \). Following Loris and Verhoeven (2011), the reference "optimal" value was computed by running the accelerated LV algorithm with bounded parameters for 100,000 iterations; this obtained the minimal value up to the point that the machine precision allows. Figure 2(a) and (d) used parameters given by Equation (22), which assumes \( x^k \) and \( y^k \) are bounded. This is true as long as \( \|x^k\|_2 < \Omega_X / \sqrt{2} \) and \( \|y^k\|_2 < \Omega_Y / \sqrt{2} \); we chose \( \Omega_X = 12 \) and \( \Omega_Y = 15 \) for group lasso, and \( \Omega_X = 141.4 \) and \( \Omega_Y = 305.9 \) for graph-guided fused lasso. The result iterates respect these bounds. Figures 2(b) and (e) used parameters given by (29), which does not require \( \Omega_X \) and \( \Omega_Y \). The oscillation in the later part of Figures 2(a) and (b) are due to the machine precision of the GPUs. Since the reference optimal value was an order of \( 10^8 \), the values in the oscillating region correspond to the 7th or 8th significant decimal digit of the objective value.

We observe that Theorems 1 and 4 faithfully describes the convergence behavior. The convergence rates of the accelerated ones were close to \( O(1/t^2) \), because in this experiment \( L_f \ll \|K\|_2 \). On the other hand, the base FB algorithms appear very close to the \( O(1/N) \) line. All of the optimal acceleration settings exhibit a very similar convergence behavior, which suggests that we have a good degree of freedom in choosing an optimal primal–dual algorithm.

Figures 2(c) and (f) compares the nonergodic convergence with respect to \( \{x^k, y^k\} \) of the FB and FBF. The FB algorithms behave like \( O(1/k) \) initially, and then converges faster than \( O(1/k^2) \). This behavior is much faster than what is predicted by Theorem 2. On the contrary, the FBF algorithm stalls after a few hundred iterations.

5.3. Scalability

To test the scalability of the studied algorithms, we consider the scenario that the number of features \( p \) is so large that, for each sample, the features do not fit into the memory. In other words, the data matrix \( A = [A^{[1]}, \ldots, A^{[M]}] \), where \( A^{[i]} \in \mathbb{R}^{n \times p_i} \), \( \sum_{i=1}^M p_i = p \), is stored distributedly in \( M \) devices. In this case, it is desirable to also split the vectors \( x \in \mathbb{R}^p \) conformally and store distributedly, that is, \( x = [x^{[1]}_1, \ldots, x^{[M]}_1]^T \), \( x_0 \in \mathbb{R}^p \). For many instances of (1) including the general lasso and group

| Table 2. Scalability of the distributed version of Equation (14) for graph-guided fused lasso and group lasso models. |
| --- |
| **Graph-guided fused lasso** |
| **#groups** | **#GPUs** | **\( p \)** |
| | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 10,000 | 120,010 | 0.019 | 4.895 | 3.801 | 2.374 | 2.468 | 2.081 | 1.739 | 1.584 | 1.518 |
| 50,000 | 600,010 | 0.023 | 20.631 | 13.779 | 11.962 | 10.124 | 8.568 | 7.699 | 6.520 |
| 80,000 | 960,010 | 0.028 | 22.695 | 16.957 | 13.712 | 11.559 | 10.343 | 10.828 |
| 100,000 | 1,200,010 | 0.031 | 20.517 | 16.190 | 15.590 | 11.704 | 12.498 |

| **Overlapping group lasso** |
| --- |
| **#groups** | **#GPUs** | **\( p \)** |
| | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 1000 | 120,010 | 0.015 | 4.826 | 4.156 | 2.973 | 2.465 | 2.102 | 1.853 | 1.591 | 1.538 |
| 5000 | 600,010 | 0.027 | 19.312 | 13.670 | 10.164 | 8.374 | 7.369 | 6.727 | 5.960 |
| 8000 | 960,010 | 0.036 | 22.792 | 17.044 | 14.722 | 12.671 | 10.866 | 10.103 |
| 10,000 | 1,200,010 | 0.049 | 22.210 | 16.658 | 15.386 | 14.088 | 11.689 |

NOTES: Time was measured in seconds per 100 iterations. Standard deviations are listed in parentheses. Any cell with missing values indicates that the experiment failed to run due to lack of memory.
6. Conclusion

In this article, we have provided a unified view to Algorithms CV and LV, two classes of primal–dual algorithms for a convex composite minimization problem based on monotone operator theory. This unification suggests a continuum of FB operator splitting algorithms for this important optimization problem having many applications in statistics. It is also this unified understanding that enables us to establish the $O(L_f/N^2 + \|K\|_2/N)$ optimal accelerations of Algorithms CV and LV (and those in between), as well as the $O(1/N)$ and $O(1/\sqrt{K})$ convergence rates for the full regions of convergence of their unaccelerated counterparts.

A practical implication of this understanding is that we bring these algorithms to the same arena: as they share the same convergence rate, other factors such as the ability of choosing wider step sizes can be fairly compared in empirical settings. Thus practitioners now possess more degrees of freedom in choosing from a suite of algorithms with theoretical guarantees.

The simplicity of the algorithms proposed and analyzed here also enables us to implement their distributed multi-GPU version almost painlessly using existing packages. This contrasts to our previous works (Yu et al. 2015; Lee et al. 2017), which resort to exploiting the structure of the matrix $K$ in Equation (1).

Supplementary material

The supplementary material contains an exposition of flexibility of formulation (1) (Appendix A), additional numerical experiments for stochastic optimal acceleration and the latent group lasso (Appendix B), a brief summary of monotone operator theory (Appendix C), and the proofs of the theorems, propositions, and lemmas (Appendix D).

Funding

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (Nos. 2019R1A2C1007126 and 2018R1C1B6001108).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015), “TensorFlow: Large-scale Machine Learning on Heterogeneous Systems,” Software available from tensorflow.org. [832]

Bauschke, H. H. and Combettes, P. L. (2011), Convex Analysis and Monotone Operator Theory in Hilbert Spaces, New York: Springer Science & Business Media. [823]

Beck, A., and Teboulle, M. (2009), “A Fast Iterative Shrinkage-thresholding Algorithm for Linear Inverse Problems,” SIAM Journal on Imaging Sciences, 2, 183–202. [822,826]

Bertsekas, D. P. (2009), Convex Optimization Theory, Belmont: Athena Scientific. [823]

Boţ, R. I., and Csetnek, E. R. (2015). “On the Convergence Rate of a Forward–Backward–Type Primal–Dual Splitting Algorithm for Convex Optimization Problems,” Optimization, 64, 5–23. [825]

Boţ, R. I., and Csetnek, E. R. (2015). “An Inertial Forward–Backward–Forward–Primal–Dual Splitting Algorithm for Solving Monotone Inclusion Problems,” Numerical Algorithms, 71, 519–540. [831]

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010), “Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers,” Foundations and Trends in Machine Learning, 3, 1–112. [822]

Chambolle, A., and Pock, T. (2011), “A First-order Primal–Dual Algorithm for Convex Problems With Applications to Imaging,” Journal of Mathematical Imaging and Vision, 40, 120–145. [822,825]

——— (2016), “On the Ergodic Convergence Rates of a First-order Primal–Dual Algorithm,” Mathematical Programming, 159, 253–287. [822,825,826]

Chen, P., Huang, J., and Zhang, X. (2013), “A Primal–Dual Fixed Point Algorithm for Convex Separable Minimization With Applications to Image Restoration,” Inverse Problems, 29, 025011. [822,824,825]

——— (2016), “A Primal–Dual Fixed Point Algorithm for Minimization of the Sum of Three Convex Separable Functions,” Fixed Point Theory and Applications, 2016, Article No. 54, 18 pp. [825]

Chen, X., Lin, Q., Kim, S., Carbonell, J. G., and Xing, E. P. (2012), “Smoothing Proximal Gradient Method for General Structured Sparse Regression,” The Annals of Applied Statistics, 6, 719–752. [822,826,829]

Chen, Y., Lan, G., and Ouyang, Y. (2014), “Optimal Primal–Dual Methods for a Class of Saddle Point Problems,” SIAM Journal on Optimization, 24, 1779–1814. [823,826,827,828,831]

Combettes, P. L., and Pesquet, J.-C. (2012), “Primal–Dual Splitting Algorithm for Solving Inclusions With Mixtures of Composite, Lipschitzian, and Parallel-Sum Type Monotone Operators,” Set-Valued and Variational Analysis, 20, 307–330. [831]

Combettes, P. L., and Wajs, V. R. (2005), “Signal Recovery by Proximal Forward–Backward Splitting,” Multiscale Modeling & Simulation, 4, 1168–1200. [822]

Combettes, P. L., Condat, L., Pesquet, J.-C., and Vu, B. C. (2014), “A Forward–Backward View of Some Primal–Dual Optimization Methods
