# Stochastic Three-Composite Convex Minimization with a Linear Operator

**Renbo Zhao**
LIONS, EPFL

**Volkan Cevher**
LIONS, EPFL

## Abstract

We develop a primal-dual convex minimization framework to solve a class of stochastic convex three-composite problem with a linear operator. We consider the cases where the problem is both convex and strongly convex and analyze the convergence of the proposed algorithm in both cases. In addition, we extend the proposed framework to deal with additional constraint sets and multiple non-smooth terms. We provide numerical evidence on graph-guided sparse logistic regression, fused lasso and overlapped group lasso, to demonstrate the superiority of our approach to the state-of-the-art.

## 1 Introduction

We study the three-composite optimization template

$$\min_{\mathbf{x}\in\mathbb{R}^d} \Big[ P(\mathbf{x}) \triangleq f(\mathbf{x}) + g(\mathbf{x}) + h(\mathbf{A}\mathbf{x}) \Big], \qquad (1)$$

where the linear operator $\mathbf{A} \in \mathbb{R}^{m\times d} \setminus \{\mathbf{0}\}$ has spectral norm $B > 0$, and $f, g : \mathbb{R}^d \to \overline{\mathbb{R}} \triangleq \mathbb{R} \cup \{\infty\}$ as well as $h : \mathbb{R}^m \to \overline{\mathbb{R}}$ are convex, closed, and proper (CCP). We assume that $f$ is continuously differentiable with $L$-Lipschitz gradient ($L>0$) on $\mathbb{R}^d$. We assume that $g$ and $h$ have tractable proximal operators.

In statistical learning, (1) can represent a doubly regularized *expected risk minimization* (ERM) problem [1]. In such cases, $f(\mathbf{x})$ assumes the following form

$$f(\mathbf{x}) := \mathbb{E}_{\boldsymbol{\xi}\sim\nu}[F(\mathbf{x}, \boldsymbol{\xi})], \qquad (2)$$

where the random variable $\boldsymbol{\xi}$ is interpreted as the data vector generated from a population distribution $\nu$, and the optimization variable $\mathbf{x}$ as the decision rule. The functions $g$ and $h$ encode regularizers or constraints.

The template (1) also has broad applications in machine learning: When $h = \iota_{\{\mathbf{b}\}}$, i.e., the indicator function function of a singleton $\{\mathbf{b}\} \subseteq \mathbb{R}^m$, (1) becomes the stochastic two-composite ERM problem with linear equality constraints [2]. Problem (1) subsumes many other important cases in machine learning, including (graph-guided) fused lasso [3, 4], constrained lasso [5], matrix completion [6] and portfolio optimization [7].

Recently, Yurtsever *et al.* [8] proposed a primal algorithm for (1) for the restricted case when $\mathbf{A} = \mathbf{I}$ based on three-operator splitting [9] . While their algorithm has wide applicability, its convergence requires strong convexity. Moreover, they cannot directly handle the non-smooth term $h$ in the template when $\mathbf{A} \neq \mathbf{I}$.

Our work directly addresses these two issues, which seem to prevalent in other relevant literature [10–12].

### 1.1 Related Works

When $h \equiv 0$, stochastic proximal gradient [13–15] algorithms have been proposed to solve (1). However, when both $g$ and $h$ are non-constant, these algorithms fail to solve (1) in general.

By disregarding the composite structure of $P$, algorithms based on stochastic subgradient [16–22] can be applied. However, the convergence of these algorithms typically rely on (i) the boundedness of the second moment of stochastic subgradients and/or (ii) the boundedness of stochastic iterates (almost surely or in expectation). For many important applications, e.g., lasso, these conditions are not satisfied on $\mathbb{R}^d$.

Some works [10–12] proposed to solve (1) using Nesterov's smoothing techniques [23]. However, these works assume that $g$ or $h$ can be written as Legendre-type transform [24] of a function with bounded domain, requiring $g$ or $h$ to be Lipschitz. Many important functions do not satisfy this assumption, e.g., the indicator functions of closed convex sets.

In [11], the authors proposed to use the proximal av-

Proceedings of the $21^{st}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. JMLR: W&CP volume 7X. 

erage techniques [25, 26] combined with (accelerated) stochastic gradient to solve (1). However, this approach requires both $g$ and $h$ to be Lipschitz, which can be rather restrictive.

By introducing a slack variable $\mathbf{y} = \mathbf{A}\mathbf{x}$, one can solve (1) via stochastic alternating direction method of multipliers (ADMM) [4,27,28]. However, similar to those stochastic subgradient methods, these works also need to assume the boundedness of stochastic subgradients and iterates. Moreover, the iterates generated by this algorithm only asymptotically satisfies the linear constraint $\mathbf{y} = \mathbf{A}\mathbf{x}$.

### 1.2 Dual and Saddle-point reformulation

Using Fenchel duality, the dual form of (1) is given by

$$\max_{\mathbf{y}\in\mathbb{R}^m} \left[ D(\mathbf{y}) \triangleq -(f+g)^*(-\mathbf{A}^T\mathbf{y}) - h^*(\mathbf{y}) \right], \quad (3)$$

where $\ell^*$ denotes the Fenchel conjugate of the function $\ell$. Similarly, the saddle-point form of (1) is given by

$$\min_{\mathbf{x}\in\mathbb{R}^d} \max_{\mathbf{y}\in\mathbb{R}^m} \big[ L(\mathbf{x},\mathbf{y}) \triangleq \mathbb{E}_{\boldsymbol{\xi}}\left[F(\mathbf{x},\boldsymbol{\xi})\right] + g(\mathbf{x}) + \langle \mathbf{A}\mathbf{x},\mathbf{y}\rangle - h^*(\mathbf{y}) \big]. \quad (4)$$

Under Slater's condition, $\mathbf{x}^*$ is an optimal solution of (1) if and only if there exists $\mathbf{y}^* \in \mathbb{R}^m$ such that $(\mathbf{x}^*, \mathbf{y}^*)$ is a saddle point of (4) [24, Theorem 36.6]. Moreover, in this case $\mathbf{y}^*$ is an optimal solution of (3).

Some special cases of (4) have been considered in previous works. Specifically, some algorithms [29–37] have been propopsed to solve (4) when $\boldsymbol{\xi}$ is deterministic. When $g \equiv 0$, Chen *et al.* [38] proposed an optimal algorithm to solve (4). Existing methods that can address the stochastic three-composite saddle-point problem (4) are subgradient-based algorithms (e.g., [16,39]). However, they cannot make use of the composite structure in (4) and are inefficient.

### 1.3 Main Contributions

We develop a primal-dual algorithm for (4) by using stochastic gradients $\{\mathbf{v}^k\}_{k\geq 0}$ that are unbiased estimators of $\nabla f$ with bounded variance. We consider two cases, i.e., when $g$ is non-strongly and strongly convex.

For non-strongly convex $g$, we consider either *constant* or *decreasing* (primal) stepsizes, depending on whether the total number of iterations $K$ is known. We show when $\mathbf{dom}\, g$ and $\mathbf{dom}\, h^*$ are bounded, the (ergodic) convergence rate of the *expected* primal-dual gap (defined in Section 3.1) is $O(1/\sqrt{K})$ with constant stepsizes and $O(\log K/\sqrt{K})$ with decreasing stepsize.

For strongly convex $g$, we consider a *decreasing* (primal) stepsize policy, regardless of the knowledge of $K$. In this case, the convergence rates of the primal-dual gap, and the squared (Euclidean) distance to the optimum, are $O(1/K)$ in expectation.

Apart from these convergence results, we also extend our proposed algorithm to the cases where i) $\mathbf{x}$ and $\mathbf{y}$ in (4) are minimized over closed convex constraint sets and ii) a *finite* number of nonsmooth terms exist in the objective function $P$.

**Notation**. We denote the Euclidean inner product by $\langle\cdot,\cdot\rangle$. Let $\|\cdot\|$ be the norm induced by $\langle\cdot,\cdot\rangle$. We use lowercase letters, bold lowercase letters and bold uppercase letters to denote scalars, vectors and matrices respectively. For any $n \geq 1$, define $[n] \triangleq \{1,\ldots,n\}$. For any $i \in [n]$, denote $\mathbf{e}_i$ as the $i$-th standard basis vector. For any CCP function $h : \mathbb{R}^m \to \overline{\mathbb{R}}$, define $\mathbf{dom}\, h \triangleq \{\mathbf{y} \in \mathbb{R}^m \,|\, h(\mathbf{y}) < \infty\}$ and for any $\mathbf{x} \in \mathbb{R}^m$,

$$\mathbf{prox}_{th}(\mathbf{x}) \triangleq \operatorname*{arg\,min}_{\mathbf{z}\in\mathbf{dom}\, h} h(\mathbf{z}) + \frac{1}{2t}\|\mathbf{x}-\mathbf{z}\|^2, \ \forall\, t > 0.$$

All the sections and lemmas with indices beginning with 'S' will appear in the supplemental material.

## 2 Algorithm

We first state some preliminaries in Section 2.1, and then provide an overview of our proposed algorithm in Section 2.2. We detail the choices of stepsizes and other parameters in Sections 2.3 and 2.4.

### 2.1 Preliminaries

We develop our algorithm to cover both cases where $P$ is non-strongly and $\gamma$-strongly convex ($\gamma > 0$). Without loss of generality, we assume that $g$ is $\gamma$-strongly convex and $f$ and $h$ are non-strongly convex.

Indeed, if $f$ is $\gamma_f$-strongly convex, we can define $\widetilde{f}_0 \triangleq f - \frac{\gamma_f}{2}\|\cdot\|^2$ and $\widetilde{g} \triangleq g + \frac{\gamma_f}{2}\|\cdot\|^2$, so that $\nabla\widetilde{f} = \nabla f - \gamma_f\|\cdot\|$ and for any $\lambda > 0$ and $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{prox}_{\lambda\widetilde{g}}(\mathbf{x}) = \mathbf{prox}_{\lambda\lambda' g}(\lambda'\mathbf{x})$, where $\lambda' \triangleq 1/(1+\lambda\gamma_f)$.

If $h$ is $\gamma_h$-strongly convex and $\mathbf{A}$ has full column rank, we can define $\widetilde{h} \triangleq h - \frac{\gamma_h}{2}\|\cdot\|^2$ and $\widetilde{f}_1 \triangleq f + \frac{\gamma_h}{2}\|\mathbf{A}\cdot\|^2$. Then, we have $\nabla\widetilde{f}_1(\mathbf{x}) = \nabla f(\mathbf{x}) + \gamma_h\mathbf{A}^T\mathbf{A}\mathbf{x}$ and for any $\lambda > \gamma_h$, $\mathbf{prox}_{\lambda\widetilde{h}^*}(\mathbf{x}) = \mathbf{x} - \lambda\mathbf{prox}_{\lambda'' h}(\lambda''\mathbf{x})$, where $\lambda'' \triangleq 1/(\lambda-\gamma_h)$. As shown in Lemma S-1, if we choose $\alpha_0 \geq 1$ and $\alpha_k$ according to (19) for any $k \geq 1$ (see below), then $\alpha_k \geq \alpha_0 \geq \gamma_h$. Hence $\mathbf{prox}_{\alpha_k\widetilde{h}^*}$ is well-defined for any $k \geq 0$ if we choose $\alpha_0 \geq \max\{1, \gamma_h\}$.

### 2.2 Overview

The pseudo-code of our algorithm is shown in Algorithm 1. Our algorithm can be regarded as a stochastic approximation of primal-dual hybrid gradient method (PDHG, also known as Chambolle-Pock) [34, 40].

Each iteration of Algorithm 1 consists of five steps. In

**Algorithm 1** Stochastic Primal-Dual Algorithm for Three-Composite Convex Minimization (SPDTCM)

1: **Input**: Positive sequences $\{\alpha_k\}_{k=0}^{K-1}$, $\{\tau_k\}_{k=0}^{K-1}$ and $\{\theta_k\}_{k=0}^{K-1}$, number of iterations $K$
2: **Initialize**: $\mathbf{x}^0 \in \mathbf{dom}\, g$, $\mathbf{y}^0 \in \mathbf{dom}\, h^*$, $\mathbf{z}^0 = \mathbf{x}^0$, $S_0 = 0$, $\overline{\mathbf{x}}^0 = \mathbf{x}^0$, $\overline{\mathbf{y}}^0 = \mathbf{y}^0$
3: **For** $k = 0, 1, \ldots, K-1$
4: Draw a sample $\boldsymbol{\xi}^k \sim \nu$ and define $\mathbf{v}^k$ as in (8)

$$\mathbf{y}^{k+1} := \mathbf{prox}_{\alpha_k h^*}(\mathbf{y}^k + \alpha_k \mathbf{A}\mathbf{z}^k) \tag{5}$$

$$\mathbf{x}^{k+1} := \mathbf{prox}_{\tau_k g}(\mathbf{x}^k - \tau_k(\mathbf{A}^T\mathbf{y}^{k+1} + \mathbf{v}^k)) \tag{6}$$

$$\mathbf{z}^{k+1} := \mathbf{x}^{k+1} + \theta_k(\mathbf{x}^{k+1} - \mathbf{x}^k) \tag{7}$$

5: **Option I**: $\beta_k = \tau_k$, **Option II**: $\beta_k = \alpha_k/\alpha_0$
6: $S_{k+1} = S_k + \beta_k$, $\overline{\mathbf{x}}^{k+1} = \left(S_k\overline{\mathbf{x}}^k + \beta_k\mathbf{x}^{k+1}\right)/S_{k+1}$
$\overline{\mathbf{y}}^{k+1} = \left(S_k\overline{\mathbf{y}}^k + \beta_k\mathbf{y}^{k+1}\right)/S_{k+1}$
7: **End for**
8: **Output**: $(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K, \mathbf{x}^K)$

the iteration $k$, we draw a sample $\boldsymbol{\xi}^k$ (independent of the past history) and obtain a stochastic gradient

$$\mathbf{v}^k \triangleq \nabla_{\mathbf{x}} F(\mathbf{x}, \boldsymbol{\xi}^k)|_{\mathbf{x}=\mathbf{x}^k}. \tag{8}$$

We then perform *dual ascent*, *primal descent* and *extrapolation* steps in (5), (6) and (7) respectively. Thus the (positive) sequences $\{\alpha_k\}_{k=0}^{K-1}$, $\{\tau_k\}_{k=0}^{K-1}$ and $\{\theta_k\}_{k=0}^{K-1}$ can be interpreted as *dual stepsizes*, *primal stepsizes* and *relaxation parameters* respectively. Note that the proximal operator $\mathbf{prox}_{\alpha_k h^*}$ in (5) can be obtained from $\mathbf{prox}_{h/\alpha_k}$ via Moreau's identity, i.e.,

$$\mathbf{prox}_{\rho h^*}(\mathbf{x}) = \mathbf{x} - \rho\mathbf{prox}_{h/\rho}(\mathbf{x}/\rho),\ \forall\, \rho > 0. \tag{9}$$

Next, we choose weight $\beta_k$ according to option I if $g$ is non-strongly convex and option II if $g$ is strongly convex. Finally, we obtain the weighted averages of iterates $\{\mathbf{x}^i\}_{i=1}^{k+1}$ and $\{\mathbf{y}^i\}_{i=1}^{k+1}$, which are denoted by $\overline{\mathbf{x}}^{k+1}$ and $\overline{\mathbf{y}}^{k+1}$ respectively. Note that $\overline{\mathbf{x}}^{k+1}$ and $\overline{\mathbf{y}}^{k+1}$ can be written explicitly as

$$\overline{\mathbf{x}}^{k+1} = \frac{\sum_{i=0}^k \beta_i \mathbf{x}^{i+1}}{S_{k+1}},\ \overline{\mathbf{y}}^{k+1} = \frac{\sum_{i=0}^k \beta_i \mathbf{y}^{i+1}}{S_{k+1}}. \tag{10}$$

For ease of presentation, in Algorithm 1 we assume to know the total number of iterations $K$ before the algorithm starts. However, in choosing the sequences $\{\alpha_k\}_{k=0}^{K-1}$, $\{\tau_k\}_{k=0}^{K-1}$ and $\{\theta_k\}_{k=0}^{K-1}$, we also consider the case where $K$ is unknown. This allows Algorithm 1 to be applied to many online and streaming applications (*cf.*, Sections 2.3 and 2.4).

### 2.3 Non-strongly convex $g$

We consider both constant and decreasing primal stepsizes $\{\tau_k\}_{k=0}^{K-1}$, depending on whether $K$ is known. As shown in Section 3.2, the constant stepsize policy can lead to slightly better convergence rates.

**Constant stepsizes ($K$ is known)**. When $K$ is known in advance, we can exploit this knowledge by choosing $\tau_k = \tau_K$, for any $0 \le k \le K-1$, where

$$\tau_K = \min\left\{\frac{\widetilde{r}}{L}, \frac{\widetilde{a}}{\widetilde{b} + \sqrt{K + \widetilde{b}'}}\right\}. \tag{11}$$

In (11), the constants $\widetilde{r}$, $\widetilde{a}$, $\widetilde{b}$ and $\widetilde{b}'$ are chosen such that $\widetilde{r} \in (0,1)$, $\widetilde{a} > 0$ and $\widetilde{b}, \widetilde{b}' \ge 0$. Our convergence results (shown in Theorem 1) hold for any values of $\widetilde{r}$, $\widetilde{a}$, $\widetilde{b}$ and $\widetilde{b}'$ that satisfy these conditions. For any $0 \le k \le K-1$, we also choose $\theta_k = 1$ and $\alpha_k = (1 - L\tau_K)/(\tau_K B^2)$.

**Decreasing stepsizes ($K$ is unknown)**. When $K$ is not known a priori, for any $k \ge 0$, we choose

$$\tau_k = \min\left\{\frac{r}{L}, \frac{a}{b + \sqrt{k + b'}}\right\}, \tag{12}$$

$$\theta_{k+1} = \frac{\tau_k}{\tau_{k+1}}, \tag{13}$$

$$\alpha_{k+1} = \frac{1 - L\tau_k}{\tau_k \theta_{k+1} B^2}. \tag{14}$$

In (12), we can choose any constants $a$, $b$, $b'$ and $r$ such that $a > 0$, $b, b' \ge 0$, $b + b' > 0$ and $r \in (0,1)$. Note that our convergence results in Theorem 1 hold for any values of $a$, $b$, $b'$ and $r$ that satisfy these conditions. Hence, we choose any $\theta_0 > 0$ and $\alpha_0 > 0$ that satisfies $\tau_1/\alpha_1 < \tau_0/\alpha_0 \le 2\tau_1/\alpha_1$. We emphasize that the constant dual stepsizes $\{\alpha_k\}_{k=0}^{K-1}$ and relaxation parameters $\{\theta_k\}_{k=0}^{K-1}$ above satisfy conditions (13) and (14).

### 2.4 Strongly convex $g$

When $g$ is $\gamma$-strongly convex, we choose any $\alpha_0 \ge 1$ and $\theta_0 > 0$, and $\{\tau_k\}_{k\ge 0}$, $\{\theta_k\}_{k\ge 1}$ and $\{\alpha_k\}_{k\ge 1}$ such that for any $k \ge 0$,

$$\theta_{k+1} = \frac{\alpha_k}{\alpha_{k+1}}, \tag{15}$$

$$\frac{1}{\tau_k} = \theta_{k+1}\alpha_{k+1}B^2 + L, \tag{16}$$

$$\frac{1}{\tau_{k+1}} = \theta_{k+1}\left(\frac{1}{\tau_k} + \gamma\right). \tag{17}$$

We provide a principled way to generate the three sequences. First, by substituting (15) into (16), we have

$$\frac{1}{\tau_k} = \alpha_k B^2 + L. \tag{18}$$

Then we substitute (15) and (18) into (17) and obtain

$$\alpha_{k+1}^2 + \frac{L}{B^2}\alpha_{k+1} = \alpha_k^2 + \frac{L}{B^2}\alpha_k + \frac{\gamma}{B^2}\alpha_k. \tag{19}$$

Thus $\alpha_{k+1}$ can be solved as the unique positive root of (19) for $\alpha_k > 0$. (Since $\alpha_0 > 0$, the whole sequence $\{\alpha_k\}_{k\ge 0}$ will be positive.) Based on $\alpha_k$ and $\alpha_{k+1}$, positive $\theta_{k+1}$, $\tau_k$ and $\tau_{k+1}$ can be generated accordingly to (15), (16) and (17).

*Remark* 1 (Scaling of parameters). In the non-strongly convex case, we can easily see that when $K$ is known, for any $0 \le k \le K-1$, $\alpha_k = \Theta(\sqrt{K})$, $\tau_k = \Theta(1/\sqrt{K})$ and $\theta_k = \Theta(1)$. When $K$ is unknown, $\alpha_k = \Theta(\sqrt{k})$, $\tau_k = \Theta(1/\sqrt{k})$ and $\theta_k = \Theta(1)$. In the strongly-convex case, as will be shown in Lemma S-1, we have $\alpha_k = \Theta(k)$, $\tau_k = \Theta(1/k)$ and $\theta_k = \Theta(1)$. The scaling of the primal stepsize $\tau_k$ agrees with that in the classical (proximal) stochastic (sub-)gradient method [14, 16, 21].

## 3 Convergence Analysis

### 3.1 Preliminaries

Given a sequence of random vectors $\{\boldsymbol{\xi}_k\}_{k\ge0}$, define $\Xi_k \triangleq \{\boldsymbol{\xi}_i\}_{i=0}^{k-1}$ for any $k \ge 1$. Accordingly, define a filtration $\{\mathcal{F}_k\}_{k\ge0}$ such that $\mathcal{F}_0 \triangleq \emptyset$ and $\mathcal{F}_k \triangleq \sigma(\Xi_k)$, i.e., the $\sigma$-field generated by $\Xi_k$.

For any set $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathbf{x} \in \mathcal{X}$, define $R_{\mathcal{X}}(\mathbf{x}) \triangleq \sup_{\mathbf{x}'\in\mathcal{X}} \|\mathbf{x}-\mathbf{x}'\|$. Define $D_g \triangleq \sup_{\mathbf{x},\mathbf{x}'\in\mathbf{dom}\, g} \|\mathbf{x}-\mathbf{x}'\|$ and $R_g(\mathbf{x}) \triangleq R_{\mathbf{dom}\, g}(\mathbf{x})$, for any $\mathbf{x} \in \mathbf{dom}\, g$. We define $D_{h^*}$ and $R_{h^*} : \mathbf{dom}\, h^* \to \mathbb{R}$ in the same way.

Following the convention in [34, 40], for any closed convex sets $\mathcal{X}' \subseteq \mathbb{R}^d$ and $\mathcal{Y}' \subseteq \mathbb{R}^m$, and any $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^m$, define the *partial primal-dual gap* $\widetilde{G}_{\mathcal{X}',\mathcal{Y}'}(\mathbf{x},\mathbf{y}) \triangleq \sup_{\mathbf{y}'\in\mathcal{Y}'} L(\mathbf{x},\mathbf{y}') - \inf_{\mathbf{x}'\in\mathcal{X}'} L(\mathbf{x}',\mathbf{y})$. Accordingly, we define the *primal-dual gap* $G(\mathbf{x},\mathbf{y}) \triangleq \widetilde{G}_{\mathbf{dom}\, g,\mathbf{dom}\, h^*}(\mathbf{x},\mathbf{y})$. Finally, for any $k \ge 0$, define the stochastic noise $\boldsymbol{\varepsilon}^k \triangleq \nabla f(\mathbf{x}^k) - \mathbf{v}^k$, where $\mathbf{v}^k$ is the stochastic gradient defined in (8).

We now state some standard (blanket) assumptions on $\{\mathbf{v}^k\}_{k\ge0}$ and $\{\boldsymbol{\varepsilon}^k\}_{k\ge0}$ below [38, 41, 42].

**Assumption 1.** *For any $k \ge 0$,*

*(a)* $\mathbb{E}_{\boldsymbol{\xi}^k}\left[\mathbf{v}^k \,|\, \mathcal{F}_k\right] = \nabla f(\mathbf{x}^k)$ *almost surely.*
*(b)* $\mathbb{E}_{\boldsymbol{\xi}^k}\left[\|\boldsymbol{\varepsilon}^k\|^2 \,\big|\, \mathcal{F}_k\right] \le \sigma^2$ *almost surely.*

We next present our convergence results. The proofs of Theorem 1, Corollaries 1 and 2 and Theorem 2 can be found in Sections S-2 to S-5 respectively.

### 3.2 Non-strongly Convex $g$

We first state a general result in Theorem 1, which applies to both constant and decreasing (primal) stepsizes $\{\tau_k\}_{k=0}^{K-1}$.

**Theorem 1.** *Let $g$ be convex. For any $K \ge 2$, choose the sequences $\{\alpha_k\}_{k=0}^{K-1}$, $\{\tau_k\}_{k=0}^{K-1}$ and $\{\theta_k\}_{k=0}^{K-1}$ such that* (13) *and* (14) *are satisfied. Use option I in Algorithm 1, then $S_K = \sum_{k=0}^{K-1}\tau_k$. Define $\widetilde{S}_K \triangleq \sum_{k=0}^{K-1}\tau_k^2$. For any bounded sets $\mathcal{X}' \subseteq \mathbb{R}^d$ and $\mathcal{Y}' \subseteq \mathbb{R}^m$ and $K \ge 2$,*

$$\mathbb{E}_{\Xi_K}\left[G_{\mathcal{X}',\mathcal{Y}'}(\overline{\mathbf{x}}^K,\overline{\mathbf{y}}^K)\right] \le \frac{1}{2S_K}\left(R^2_{\mathcal{X}'}(\mathbf{x}^0) + \frac{\tau_0}{\alpha_0}R^2_{\mathcal{Y}'}(\mathbf{y}^0)\right) + \frac{\widetilde{S}_K}{S_K}\sigma^2.$$

*In particular, if* $\mathbf{dom}\, g$ *and* $\mathbf{dom}\, h^*$ *are bounded, then*

$$\mathbb{E}_{\Xi_K}\left[G(\overline{\mathbf{x}}^K,\overline{\mathbf{y}}^K)\right] \le \frac{1}{2S_K}\left(R^2_g(\mathbf{x}^0) + \frac{\tau_0}{\alpha_0}R^2_{h^*}(\mathbf{y}^0)\right) + \frac{\widetilde{S}_K}{S_K}\sigma^2. \quad (20)$$

*Remark* 2. Note that the boundedness of $\mathbf{dom}\, h^*$ is equivalent to the Lipschitz continuity of $h$ on $\mathbb{R}^m$. Many important nonsmooth functions are Lipschitz, such as norm and Huber loss [43]. If $g$ involves an indicator function of a convex and compact set, then $\mathbf{dom}\, g$ is bounded. As will be discussed in Section 4.1, if (4) is minimized over bounded sets $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}^m$, then the boundedness requirements on $\mathbf{dom}\, g$ and $\mathbf{dom}\, h^*$ can be removed.

**Implications of Theorem 1**. Theorem 1 shows that the convergence rate of the expected primal-dual gap in (20) scales as $\Theta(\widetilde{S}_K/S_K)$. However, $\widetilde{S}_K/S_K$ has different scalings depending on whether $K$ is known, due to the choice of $\{\alpha_k\}_{k=0}^{K-1}$ (cf. Section 2.3).

Recall from Remark 1 that $\tau_k = \Theta(1/\sqrt{K})$ and $\tau_k = \Theta(1/\sqrt{k})$ in the $K$-known and $K$-unknown cases respectively. Thus $S_K = \Theta(\sqrt{K})$ in both cases. However, $\widetilde{S}_K = \Theta(1)$ when $K$ is known and $\widetilde{S}_K = \Theta(\log K)$ otherwise. As a result, $\widetilde{S}_K/S_K = \Theta(1/\sqrt{K})$ and $\widetilde{S}_K/S_K = \Theta(\log K/\sqrt{K})$ when $K$ is known and unknown respectively.

Based on Theorem 1, by judiciously choosing the constants in the primal stepsizes $\{\tau_k\}_{k=0}^{K-1}$ in (11) and (12) respectively, we have the following two corollaries.

**Corollary 1.** *Let $g$ be convex and* $\mathbf{dom}\, g$ *and* $\mathbf{dom}\, h^*$ *be bounded. Use option I in Algorithm 1. In* (11)*, let*

$$\widetilde{a} = \frac{R_g(\mathbf{x}^0)}{\sqrt{3}\sigma}, \quad \widetilde{b} = 0 \quad \text{and} \quad \widetilde{b}' = \frac{B^2R^2_{h^*}(\mathbf{y}^0)}{3\sigma^2(1-\widetilde{r})}. \quad (21)$$

*For any $K \ge 1$ and $\widetilde{r} \in (0,1)$, if*

$$L \ge \frac{\widetilde{r}}{R_g(\mathbf{x}^0)}\sqrt{\frac{B^2R^2_{h^*}(\mathbf{y}^0)}{1-\widetilde{r}} + 3K\sigma^2}, \quad (22)$$

*then*

$$\mathbb{E}_{\Xi_K}\left[G(\overline{\mathbf{x}}^K,\overline{\mathbf{y}}^K)\right] \le \frac{R^2_g(\mathbf{x}^0)L}{2K\widetilde{r}} + \frac{R_g(\mathbf{x}^0)R_{h^*}(\mathbf{y}^0)B}{2K\sqrt{1-\widetilde{r}}} + \frac{\sqrt{3}R_g(\mathbf{x}^0)\sigma}{2\sqrt{K}}. \quad (23)$$

*Otherwise,*

$$\mathbb{E}_{\Xi_K}\left[G(\overline{\mathbf{x}}^K,\overline{\mathbf{y}}^K)\right] \le \frac{R_g(\mathbf{x}^0)R_{h^*}(\mathbf{y}^0)B}{K\sqrt{1-\widetilde{r}}} + \frac{\sqrt{3}R_g(\mathbf{x}^0)\sigma}{\sqrt{K}}.$$

*Remark* 3. We note that the smooth function $f$ (represented by the first terms in (23)) contributes to the

convergence rate of the primal-dual gap only when $L = \Omega(\sqrt{K})$. This squares our intuition since when $f$ is sufficiently smooth, i.e., $L = O(\sqrt{K})$, the cost of minimizing it should be outweighed by other components. See [18, Remark 1] for a detailed discussion.

*Remark* 4. From Corollary 1, we observe that the expected primal-dual gap $G(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K)$ converges at $O(L/K + B/K + \sigma/\sqrt{K})$. When $g \equiv 0$, a lower bound for this convergence rate for any stochastic first-order algorithm was shown to be $\Omega(L/K^2 + B/K + \sigma/\sqrt{K})$ in [38, Section 1.2]. Compared to this lower bound, we notice that the convergence rates of the bilinear term $\langle \mathbf{Ax}, \mathbf{y} \rangle$ and the stochastic noise (represented by $B/K$ and $\sigma/\sqrt{K}$ respectively) are indeed optimal. However, the convergence rate for the smooth part $f$ (represented by $L/K$) may be improved to $O(L/K^2)$ using acceleration [18,38]. We defer details to future work.

**Corollary 2.** *Let $g$ be convex. In* (12)*, choose $a = \widetilde{a}$, $b = \widetilde{b}$, $b' = \widetilde{b}' + 1$ and $r = \widetilde{r}$ as in Corollary 1 and use option I in Algorithm 1. Let* $\mathbf{dom}\, g$ *and* $\mathbf{dom}\, h^*$ *be both bounded. For any $K \geq 2$, define*

$$C_K \triangleq \left(1 + \frac{1}{3\sigma^2}\right) \frac{R_g^2(\mathbf{x}^0)(1-\widetilde{r})}{B^2 R_{h^*}^2(\mathbf{y}^0)} + \frac{R_g^2(\mathbf{x}^0)}{3\sigma^2} \log\left(\frac{B^2 R_{h^*}^2(\mathbf{y}^0)}{1-\widetilde{r}} + 3\sigma^2 K\right) = O(\log K).$$

*For any $K \geq 2$ and $\widetilde{r} \in (0,1)$, if* (22) *holds, then*

$$\mathbb{E}_{\Xi_K}\left[G(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K)\right] \leq \frac{R_g^2(\mathbf{x}^0) L}{2K\widetilde{r}} + \frac{R_g(\mathbf{x}^0) R_{h^*}(\mathbf{y}^0) B}{2K\sqrt{1-\widetilde{r}}} + \frac{\sqrt{3} R_g(\mathbf{x}^0)\sigma}{2\sqrt{K}}. \tag{24}$$

*Otherwise,*

$$\mathbb{E}_{\Xi_K}\left[G(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K)\right] \leq \left(\frac{3R_g^2(\mathbf{x}^0)}{2} + \sigma^2 C_K\right)\left(\frac{B R_{h^*}(\mathbf{y}^0)}{K\sqrt{1-\widetilde{r}}} + \frac{\sqrt{3}\sigma}{\sqrt{K}}\right). \tag{25}$$

*Remark* 5. Corollary 2 suggests that the primal-dual gap $G(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K)$ converges at $O(\log K/\sqrt{K})$, which does not match the lower bound stated in Remark 4. One possible approach to eliminate the $\log K$ factor was introduced in [16, Section 2.2]. Specifically, define $\underline{K} \triangleq \lceil \upsilon K \rceil$, where $\upsilon \in (0,1)$ is independent of $K$. In Algorithm 1, $\overline{\mathbf{x}}^K$ and $\overline{\mathbf{y}}^K$ are generated by averaging $\{\mathbf{x}^k\}_{k=\underline{K}}^K$ and $\{\mathbf{y}^k\}_{k=\underline{K}}^K$ respectively. However, since $K$ is unknown, this approach requires us to store all the iterates $\{\mathbf{x}^k\}_{k=1}^K$ and $\{\mathbf{y}^k\}_{k=1}^K$, and hence becomes impractical when memory is limited.

### 3.3 Strongly Convex $g$

**Theorem 2.** *Let $g$ be $\gamma$-strongly convex. Use option II in Algorithm 1. Choose the sequences $\{\alpha_k\}_{k=0}^{K-1}$, $\{\tau_k\}_{k=0}^{K-1}$ and $\{\theta_k\}_{k=0}^{K-1}$ as in Section 2.4. Define two constants $\overline{c}_1 \triangleq (\alpha_0 B^2 + \gamma)(2B^2 + 2L + \gamma)/(\alpha_0 \gamma B^4)$ and $\overline{c}_1' \triangleq \max\left\{4\alpha_0(2B^2 + 2L + \gamma)/\gamma, 1\right\}$ Then for any $K \geq 1$ and any bounded sets $\mathcal{X}' \subseteq \mathbb{R}^d$ and $\mathcal{Y}' \subseteq \mathbb{R}^m$,*

$$\mathbb{E}_{\Xi_K}\left[G_{\mathcal{X}',\mathcal{Y}'}\left(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K\right)\right] \leq \frac{\overline{c}_1'}{2K^2}\left(\frac{1}{\tau_0} R_{\mathcal{X}'}^2(\mathbf{x}^0) + \frac{1}{\alpha_0} R_{\mathcal{Y}'}^2(\mathbf{y}^0)\right) + \frac{\overline{c}_1 \overline{c}_1' \sigma^2}{K}. \tag{26}$$

*In particular, if* $\mathbf{dom}\, g$ *and* $\mathbf{dom}\, h^*$ *are bounded, then*

$$\mathbb{E}_{\Xi_K}\left[G\left(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K\right)\right] \leq \frac{\overline{c}_1'}{2K^2}\left(\frac{1}{\tau_0} R_g^2(\mathbf{x}^0) + \frac{1}{\alpha_0} R_{h^*}^2(\mathbf{y}^0)\right) + \frac{\overline{c}_1 \overline{c}_1' \sigma^2}{K}. \tag{27}$$

*(ii) Denote the unique minimizer of* (1) *by $\mathbf{x}^*$. Define $c_2 \triangleq (2B^2 + 2L + \gamma)^2/(B^2\gamma^2)$. Then for any $K \geq 1$, there exists $\mathbf{y}^* \in \mathbb{R}^m$ such that*

$$\mathbb{E}_{\Xi_K}\left[\|\mathbf{x}^K - \mathbf{x}^*\|^2\right] \leq \frac{c_2 \alpha_0}{K^2}\left(\frac{\|\mathbf{x}^0 - \mathbf{x}^*\|^2}{\tau_0} + \frac{\|\mathbf{y}^0 - \mathbf{y}^*\|^2}{\alpha_0}\right) + \frac{2\overline{c}_1 c_2 \alpha_0 \sigma^2}{K}. \tag{28}$$

*Remark* 6. Note that for the convergence results concerning $\|\mathbf{x}^K - \mathbf{x}^*\|^2$ in part (ii), we assume neither the boundedness of $\mathbf{dom}\, g$ or $\mathbf{dom}\, h^*$, nor the uniform boundedness of $\partial P$ on $\mathbb{R}^d$. This distinguishes our work from many stochastic algorithms in the literature that are based on subgradient [21,22,44], Nesterov's smoothing [10–12] or ADMM [4, 27, 28]. In fact, without the aforementioned assumptions, it is unclear whether these algorithms even converge when $P$ is strongly convex. In contrast, we show an $O(1/K)$ convergence rate of $\|\mathbf{x}^K - \mathbf{x}^*\|^2$ in expectation.

*Remark* 7. We observe that the convergence rates in both (27) and (28) depend on $\sigma$ by $O(\sigma^2/K)$. This dependence is indeed optimal, since when $h \equiv 0$, for any stochastic first-order algorithm, a lower bound was shown to be $\Omega(\sigma^2/K)$ in [19, Section 1].

## 4 Extensions

### 4.1 Constrained Saddle-Point Problems

Let $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}^m$ be two closed convex sets. Then the constrained version of problem (4) is

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \mathbb{E}_{\boldsymbol{\xi}}\left[F(\mathbf{x}, \boldsymbol{\xi})\right] + g(\mathbf{x}) + \langle \mathbf{Ax}, \mathbf{y} \rangle - h^*(\mathbf{y}). \tag{29}$$

Indeed, to extend Algorithm 1 to solve (29), we only need to change the dual step (5) and the primal

step (6) in Algorithm 1 to

$$\mathbf{y}^{k+1} = \arg\min_{\mathbf{y}\in\mathcal{Y}} \frac{\|\mathbf{y}-\mathbf{y}^k\|^2}{2\alpha_k} - \langle \mathbf{y}, \mathbf{A}\mathbf{z}^k\rangle + h^*(\mathbf{y}), \quad (30)$$

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \frac{\|\mathbf{x}-\mathbf{x}^k\|^2}{2\tau_k} + \langle \mathbf{x}, \mathbf{A}^T\mathbf{y}^{k+1} + \mathbf{v}^k\rangle + g(\mathbf{x}). \quad (31)$$

All of the convergence results in Section 3 apply to the new algorithm for solving (29). In particular, to prove the convergence results for the primal-dual gap, we only need to assume that $\mathbf{dom}\, g \cap \mathcal{X}$ and $\mathbf{dom}\, h^* \cap \mathcal{Y}$ are bounded. This condition is certainly true, if both $\mathcal{X}$ and $\mathcal{Y}$ are compact, which has been assumed in many previous works [10, 16, 45].

*Remark* 8. Note that the steps (30) or (31) may not be solved in closed form or finitely many steps, even if $\mathbf{prox}_{\alpha_k h^*}$ or $\mathbf{prox}_{\tau_k g}$ does. In this case, one needs to conduct *inexact* analysis by accounting for the computational errors associated with solving (30) and (31). We leave such an analysis to future work.

### 4.2 Multiple Nonsmooth Terms

We consider a stochastic convex minimization problem with $p+1$ nonsmooth terms ($p \geq 1$), i.e.,

$$\min_{\mathbf{x}\in\mathbb{R}^d} \mathbb{E}_{\boldsymbol{\xi}\sim\nu}[F(\mathbf{x},\boldsymbol{\xi})] + g(\mathbf{x}) + \sum_{i=1}^p r_i(\mathbf{A}_i\mathbf{x}), \quad (32)$$

where $\{r_i\}_{i=1}^p$ are proper, closed, convex functions such that each $r_i$ has a tractable proximal operator, and $\mathbf{A}_i \in \mathbb{R}^{m\times d}$ for any $i \in [p]$. For simplicity, we let all the matrices $\{\mathbf{A}_i\}_{i=1}^p$ have the same number of rows and leave the simple generalization, where the numbers of rows of $\{\mathbf{A}_i\}_{i=1}^p$ are different, to the reader.

The problem (32) has numerous applications, including overlapping group lasso [46], robust matrix recovery [47] and variational image recovery [48]. When $\boldsymbol{\xi}$ is deterministic, numerous algorithms have been developed to solve (32), such as [29–32, 48]. However, when $\boldsymbol{\xi}$ has a general probability distribution, so far there exists no method in the literature that can solve (32).

We next describe how to reformulate (32) in the form of (1) and develop an algorithm to solve (32) based on Algorithm 1. For any vector $\dot{\mathbf{x}} \in \mathbb{R}^{pd}$, denote its $i$-th block as $\dot{\mathbf{x}}_i \triangleq (x_{p(i-1)+1}, \ldots, x_{pi})^T \in \mathbb{R}^d$, where $i \in [p]$. We also denote $\dot{\mathbf{y}} \in \mathbb{R}^{pm}$ and $\{\dot{\mathbf{y}}_i\}_{i=1}^p \subseteq \mathbb{R}^m$ in the same way. We define $\mathcal{V} \triangleq \{\dot{\mathbf{x}} \in \mathbb{R}^{pd} \,|\, \dot{\mathbf{x}}_1 = \dot{\mathbf{x}}_2 = \cdots = \dot{\mathbf{x}}_p\}$. Recall that $f(\cdot) = \mathbb{E}_{\boldsymbol{\xi}}[F(\cdot,\boldsymbol{\xi})]$ in (2). To extend Algorithm 1 to solve (32), we first rewrite (32) as a (stochastic) three-composite minimization problem in the augmented space $\mathbb{R}^{pd}$, i.e.,

$$\min_{\dot{\mathbf{x}}\in\mathbb{R}^{pd}} \left[ \dot{P}(\dot{\mathbf{x}}) \triangleq \frac{1}{p}\sum_{i=1}^p f(\dot{\mathbf{x}}_i) + g(\dot{\mathbf{x}}_i) + \iota_{\mathcal{V}}(\dot{\mathbf{x}}) + \sum_{i=1}^p r_i(\mathbf{A}_i\dot{\mathbf{x}}_i) \right]. \quad (33)$$

An equivalent saddle-point form of (33) is

$$\min_{\dot{\mathbf{x}}\in\mathbb{R}^{pd}} \max_{\dot{\mathbf{y}}\in\mathbb{R}^{pm}} \left[ \dot{L}(\dot{\mathbf{x}},\dot{\mathbf{y}}) \triangleq \frac{1}{p}\sum_{i=1}^p f(\dot{\mathbf{x}}_i) + g(\dot{\mathbf{x}}_i) + \iota_{\mathcal{V}}(\dot{\mathbf{x}}) + \sum_{i=1}^p \langle \mathbf{A}_i\dot{\mathbf{x}}_i, \dot{\mathbf{y}}_i\rangle - \sum_{i=1}^p r_i^*(\dot{\mathbf{y}}_i) \right]. \quad (34)$$

For convenience, define $J(\dot{\mathbf{x}}) \triangleq (1/p)\sum_{i=1}^p f(\dot{\mathbf{x}}_i)$, $Q(\dot{\mathbf{x}}) \triangleq (1/p)\sum_{i=1}^p g(\dot{\mathbf{x}}_i)$ and $R^*(\dot{\mathbf{y}}) \triangleq \sum_{i=1}^p r_i^*(\dot{\mathbf{y}}_i)$. Then for any $i \in [p]$, $(\nabla J(\dot{\mathbf{x}}))_i = (1/p)\nabla f(\dot{\mathbf{x}}_i)$, $(\mathbf{prox}_{Q+\iota_{\mathcal{V}}}(\dot{\mathbf{x}}))_i = \mathbf{prox}_{g/p}(\frac{1}{p}\sum_{i=1}^p \dot{\mathbf{x}}_i)$ and for any $\lambda > 0$, $(\mathbf{prox}_{\lambda R^*}(\dot{\mathbf{y}}))_i = \mathbf{prox}_{\lambda r_i^*}(\dot{\mathbf{y}}_i)$. Based on above, we develop an algorithm to solve (34) in Algorithm 2.

Note that Algorithm 2 admits an efficient parallel implementation. Specifically, the computationally intensive steps (36), (37) and (38) can be performed simultaneously across $p$ nodes. In terms of storage, at iteration $k$, each node only needs to store the latest average vector $\overline{\mathbf{y}}_i^{k+1} \in \mathbb{R}^m$, and the central node only needs to store $\overline{\mathbf{x}}^{k+1} \in \mathbb{R}^d$.

Depending on the convexity of $g$, and the knowledge of $K$, we can choose $\{\alpha_k\}_{k\geq 0}$, $\{\tau_k\}_{k\geq 0}$, $\{\theta_k\}_{k\geq 0}$ and $\beta_k$ in a similar fashion as in Section 2. Based on the convergence results of Algorithm 1 (shown in Theorems 1 and 2) in the augmented spaces $\mathbb{R}^{pd}$ and $\mathbb{R}^{pm}$, we can obtain *all* the corresponding convergence results of Algorithm 2 in $\mathbb{R}^d$ and $\mathbb{R}^m$ in a straightforward manner. To be specific, we provide an example below.

**Corollary 3.** *Let $g$ be $\gamma$-strongly convex. Choose the sequences $\{\alpha_k\}_{k\geq 0}$, $\{\tau_k\}_{k\geq 0}$ and $\{\theta_k\}_{k\geq 0}$ as in Section 2.4 and Option II in Algorithm 2. Define $\dot{c}_1 \triangleq (\alpha_0\dot{B}^2+\gamma)(2\dot{B}^2+2L+\gamma)/(\gamma\dot{B}^4)$, $\dot{c}_2 \triangleq (2\dot{B}^2+2L+\gamma)^2/(\dot{B}^2\gamma^2)$ and $\dot{B} \triangleq \max_{i=1}^p \|\mathbf{A}_i\|_2$. Denote the unique minimizer of (32) by $\widetilde{\mathbf{x}}^*$. Then for any $K \geq 1$, there exist $\{\widetilde{\mathbf{y}}_1^*, \ldots, \widetilde{\mathbf{y}}_p^*\} \subseteq \mathbb{R}^m$ such that*

$$\mathbb{E}_{\Xi_K}\left[\|\mathbf{x}^K - \widetilde{\mathbf{x}}^*\|^2\right] \leq \frac{\dot{c}_2}{K^2}\left(\frac{\alpha_0}{\tau_0}\|\mathbf{x}^0 - \widetilde{\mathbf{x}}^*\|^2 + \frac{1}{p}\sum_{i=1}^p \|\mathbf{y}_i^0 - \widetilde{\mathbf{y}}_i^*\|^2\right) + \frac{2\dot{c}_1\dot{c}_2\sigma^2}{K}. \quad (35)$$

## 5 Applications and Experiments

### 5.1 Experimental Setup

**Benchmark Algorithms**. The benchmark algorithms including one batch (deterministic) algorithm and two stochastic algorithms. The deterministic method is based on PDHG [34]. The two stochastic

**Algorithm 2** Stochastic Primal-Dual Algorithm for Multi-Composite Convex Minimization (SPDMCM)

**Input**: Positive sequences $\{\alpha_k\}_{k\geq 0}$, $\{\tau_k\}_{k\geq 0}$ and $\{\theta_k\}_{k\geq 0}$, weights $\{w_i\}_{i=1}^p$, number of iterations $K$
**Initialize**: $\mathbf{x}^0 \in \mathbb{R}^d$, $\mathbf{y}_1^0, \cdots, \mathbf{y}_p^0 \in \mathbb{R}^m$, $\mathbf{z}^0 = \mathbf{x}^0$
**For** $k = 0, 1, \ldots, K-1$
Draw a sample $\boldsymbol{\xi}^k \sim \nu$ and define $\mathbf{v}^k$ as in (8)
**Option I**: $\beta_k = \tau_k$, **Option II**: $\beta_k = \alpha_k/\alpha_0$
$S_{k+1} := S_k + \beta_k$
For each $i \in [p]$, perform steps (36), (37) and (38) (in parallel)

$$\mathbf{y}_i^{k+1} := \mathbf{prox}_{\alpha_k r_i^*}(\mathbf{y}_i^k + \alpha_k \mathbf{A}_i \mathbf{z}^k) \tag{36}$$

$$\boldsymbol{\zeta}_i^{k+1} := \mathbf{x}^k - \tau_k(\mathbf{A}_i^T \mathbf{y}_i^{k+1} + (1/p)\mathbf{v}^k) \tag{37}$$

$$\overline{\mathbf{y}}_i^{k+1} := \left(S_k \overline{\mathbf{y}}_i^k + \beta_k \mathbf{y}_i^{k+1}\right)/S_{k+1} \tag{38}$$

$\mathbf{x}^{k+1} := \mathbf{prox}_{g/p}\left(\frac{1}{p}\sum_{i=1}^p \boldsymbol{\zeta}_i^{k+1}\right)$
$\overline{\mathbf{x}}^{k+1} := \left(S_k \overline{\mathbf{x}}^k + \beta_k \mathbf{x}^{k+1}\right)/S_{k+1}$
$\mathbf{z}^{k+1} := \mathbf{x}^{k+1} + \theta_k(\mathbf{x}^{k+1} - \mathbf{x}^k)$
**End for**
**Output**: $\overline{\mathbf{x}}^K$, $\{\overline{\mathbf{y}}_i^K\}_{i=1}^p$ and $\mathbf{x}^K$

methods are based on stochastic smoothing combined with proximal average [11] and stochastic ADMM [4, 28] respectively. We denote the three benchmark algorithms as PDHG, SSPA and SADMM respectively.

In our comparison, we set the parameter values in these benchmark algorithms according to the original papers. In addition, we repeated each stochastic algorithm (including ours) ten times from the same starting point and show the average realization.

**Parameter Choices**. For non-strongly convex $P$, for known $K$, we set $\widetilde{r} = 0.3$, $\widetilde{a} = 100$, $\widetilde{b} = 0$ and $\widetilde{b'} = 1$ in (11). For unknown $K$, we used the same parameters in the stepsize $\tau_k$, i.e., we chose $r = \widetilde{r}$, $a = \widetilde{a}$, $b = \widetilde{b}$ and $b' = \widetilde{b'}$ in (12). We also chose $\theta_0 = 1$ and $\alpha_0 = \tau_0\alpha_1/(2\tau_1)$. For strongly convex $P$, we chose $\alpha_0 = 0.5$ and $\theta_0 = 1$. These simple parameter choices indeed worked well in our experiments.

**Tasks and Datasets**. We considered three regression problems, which are graph-guided sparse logistic regression [49], fused lasso [3] and overlapped group lasso [46]. The datasets were extracted from the LIBSVM [50] repository. Each dataset $\mathcal{D}$ of size $n$ can be represented as $\{(\mathbf{a}_i, b_i)\}_{i=1}^n$, where $\{\mathbf{a}_i\}_{i=1}^n \subseteq \mathbb{R}^d$ denote the feature vectors and $\{b_i\}_{i=1}^n \subseteq \mathbb{R}$ denote the response variables. Accordingly, define the data matrix $\mathbf{A} \triangleq [\mathbf{a}_1 \ldots \mathbf{a}_n]^T$.

**Comparison Criteria**. Since the theoretical convergence rates of our algorithm and the benchmark ones are given in terms of the average iterates, we use the (empirical) *ergodic* primal suboptimality, i.e., $P(\overline{\mathbf{x}}^k) - P^*$, to compare all the algorithms. We term $n$ data samples as one *epoch*.

We plot the decrease of the primal suboptimality versus both the number of epochs and time (in seconds). Both plots have pros and cons. Specifically, the epoch-plot cannot reflect the amount of computation to utilize each data sample whereas the time-plot is highly dependent on implementation. Therefore, we believe a cross-reference of both plots provides a more comprehensive view on the method efficiency.

All the algorithms were implemented in Matlab® R2016b on a machine with 1.7 GHz Intel® i5-4210U processor and 8 GB RAM.

### 5.2 Graph-Guided Sparse Logistic Regression

For any $i \in [n]$, define

$$\ell_i(\mathbf{x}) \triangleq \log\left(1 + \exp\left(-b_i \mathbf{a}_i^T \mathbf{x}\right)\right). \tag{39}$$

The graph-guided sparse logistic regression is given by

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[ P_{\mathrm{LR}}(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^n \ell_i(\mathbf{x}) + \lambda_1 \|\mathbf{x}\|_1 + \lambda_2 \|\mathbf{F}\mathbf{x}\|_1 \right],$$

where $\lambda_1$ and $\lambda_2$ are positive regularization parameters and $\mathbf{F}$ is a matrix that encodes the fusion penalty [3].

We set $\lambda_1 = \lambda_2 = 1$. Note that $P_{\mathrm{LR}}$ is non-strongly convex on $\mathbb{R}^d$, for any dataset $\mathcal{D}$. The smoothness parameter $L = \sigma_{\max}^2(\mathbf{A})/(4n)$. We obtained the matrix $\mathbf{F}$ in a similar fashion as in [4]. At iteration $k$, to form the stochastic gradient $\mathbf{v}^k$, we first uniformly randomly sampled an index set $\mathcal{B}_k \subseteq [n]$ without replacement, such that $|\mathcal{B}_k| = \lfloor 0.01n \rfloor$. Then we let

$$\mathbf{v}^k = \frac{1}{|\mathcal{B}_k|}\sum_{i\in\mathcal{B}_k} \nabla \ell_i(\mathbf{x}^k). \tag{40}$$

This procedure was also used in the fused lasso and overlapped group lasso tasks (see Sections 5.3 and 5.4).

We tested the performance of Algorithm 1 against the benchmark algorithms on the `a9a` and `w8a` datasets. We implemented Algorithm 1 with both decreasing and constant (primal) stepsizes, which are denoted by Ours(Dec) and Ours(Cst) respectively.

The results are shown in Figures 1 and 2. We observe that our algorithm, with both decreasing and constant stepsizes, outperforms all the benchmark methods, in terms of both epochs and time. The reason that both stepsizes perform similarly is because when $K$ is not large enough, the same stepsize $\widetilde{r}/L$ is used.

In addition, in terms of time, SSPA and SADMM perform similarly compared to PDHG. This is because: i) Each iteration in the SSPA and SADMM algorithms requires computing the proximal average of $\|\mathbf{F}\mathbf{x}\|_1$ and $\mathbf{F}^T\mathbf{F}$ respectively, which can be expensive; ii) To use

the same number of data samples, the stochastic algorithms (including ours) requires much more (indeed, 100 times) iterations than PDHG. The overhead in the `for` or `while` loops also contributes to the slowdown. However, even with the overhead, our algorithm still outperforms PDHG by a large margin (in time).

### 5.3 Fused Lasso

Define a matrix $\mathbf{D} \in \mathbb{R}^{(d-1)\times d}$ such that for any $i \in [d-1]$, $D_{i,i} = 1$ and $D_{i,i+1} = -1$. All of its remaining entries are zero. Also define

$$\widetilde{\ell}_i(\mathbf{x}) \triangleq \frac{1}{2}\left(\mathbf{a}_i^T\mathbf{x} - b_i\right)^2, \ \forall\, i \in [n]. \tag{41}$$

We then formulate the fused lasso problem as

$$\min_{\mathbf{x}\in\mathbb{R}^d}\left[P_{\mathrm{FL}}(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^n \widetilde{\ell}_i(\mathbf{x}) + \lambda_1\|\mathbf{x}\|_1 + \lambda_2\|\mathbf{D}\mathbf{x}\|_1\right],$$

the regularization weights $\lambda_1 = \lambda_2 = 1$. The smoothness parameter $L = \sigma_{\max}^2(\mathbf{A})/n$. If $\mathbf{A}$ has full column-rank, then $P_{\mathrm{FL}}$ is strongly convex with modulus $\gamma = \sigma_{\min}^2(\mathbf{A})/n > 0$.

We tested the performance of all the algorithms on the `YearPrediction` dataset, whose data matrix $\mathbf{A}$ has full column rank. The results are shown in Figure 3. We indeed have similar observations to those in Section 5.2. In particular, our algorithm converges faster compared to all the benchmark algorithms, in terms of the number of epochs and running time.

### 5.4 Overlapping Group Lasso

We generated $\widetilde{p}$ groups of indices from $[d]$, each of size $q$, by random sampling without replacement. These groups were denoted by $\{\widetilde{\mathcal{G}}_i\}_{i=1}^{\widetilde{p}}$. We partitioned $[\widetilde{p}]$ into $\{\mathcal{I}_{i'}\}_{i'=1}^{p}$ ($p \le \widetilde{p}$), such that for any $i' \in [p]$ and any $j, j' \in \mathcal{I}_{i'}$, $\widetilde{\mathcal{G}}_j \cap \widetilde{\mathcal{G}}_{j'} = \emptyset$. Accordingly, for any $i' \in [p]$ and $\mathbf{x} \in \mathbb{R}^d$, define $\mathcal{G}_{i'} \triangleq \bigcup_{j\in\mathcal{I}_{i'}} \widetilde{\mathcal{G}}_j$ and $\mathbf{x}_{\mathcal{G}_{i'}}$ to be the subvector of $\mathbf{x}$ with indices from $\mathcal{G}_{i'}$. Based on these notations, the overlapped group lasso is formulated as

$$\min_{\mathbf{x}\in\mathbb{R}^d}\left[P_{\mathrm{GL}}(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^n \widetilde{\ell}_i(\mathbf{x}) + \sum_{i'=1}^p \lambda_{i'}\left\|\mathbf{x}_{\mathcal{G}_{i'}}\right\|\right]. \tag{42}$$

Similar to previous tasks, we set $\lambda_{i'} = 1$, for any $i' \in [p]$. Let $\pi_{i'} : [|\mathcal{G}_{i'}|] \to \mathcal{G}_{i'}$ be any bijective map. We note that the problem (42) fits into the template (32), since $\mathbf{x}_{\mathcal{G}_{i'}} = \mathbf{U}_{i'}\mathbf{x}$, where $\mathbf{U}_{i'} \in \mathbb{R}^{|\mathcal{G}_{i'}|\times d}$ and $(\mathbf{U}_{i'})_{j:} = \mathbf{e}_{\pi_{i'}(j)}$. As a result, we apply Algorithm 2 to solve it.

We tested all the algorithms on the `E2006-tfidf` dataset. For simplicity, we subsampled 1000 features with the highest frequencies. The resulting data matrix $\mathbf{A}$ has full column rank so $P_{\mathrm{GL}}$ is strongly convex.

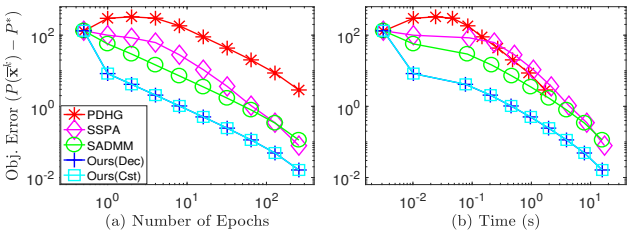


Figure 1: Loglog plot of the objective error $P(\overline{\mathbf{x}}^k) - P^*$ versus (a) number of epochs and (b) time (in seconds) on the `a9a` dataset.

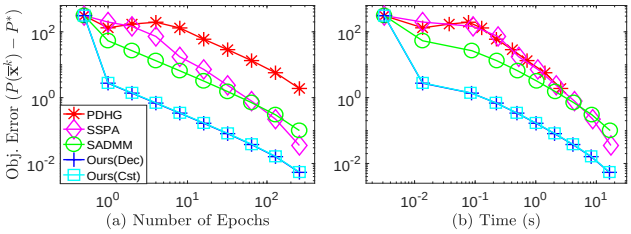


Figure 2: Loglog plot of the objective error $P(\overline{\mathbf{x}}^k) - P^*$ versus (a) number of epochs and (b) time (in seconds) on the `w8a` dataset.

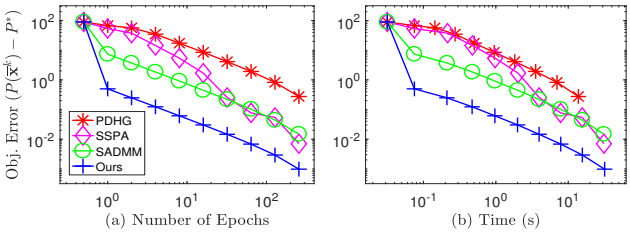


Figure 3: Loglog plot of the objective error $P(\overline{\mathbf{x}}^k) - P^*$ versus (a) number of epochs and (b) time (in seconds) on the `YearPrediction` dataset.

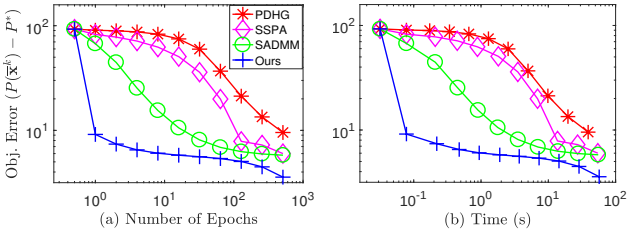


Figure 4: Loglog plot of the objective error $P(\overline{\mathbf{x}}^k) - P^*$ versus (a) number of epochs and (b) time (in seconds) on the `E2006-tfidf` dataset.

We set $p = 5$ and $q = \lfloor 0.3d \rfloor$. The results are shown in Figure 4. We observe that in terms of both the number of epochs and running time, our algorithm convergences faster than all the benchmark algorithms.

**Acknowledgement**. This work has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program (grant agreement No. 725594 – time-data).

## References


[1] V. N. Vapnik, *The nature of statistical learning theory*. Springer-Verlag New York, 2000.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[3] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. R. Stat. Soc. Ser. B*, vol. 67, no. 1, pp. 91–108, 2005.

[4] H. Ouyang, N. He, L. Tran, and A. Gray, "Stochastic alternating direction method of multipliers," in *Proc. ICML*, (Atlanta, USA), pp. 80–88, 2013.

[5] G. M. James, C. Paulson, and P. Rusmevichientong, "The constrained lasso," tech. rep., University of Southern California, 2013.

[6] T. T. Cai and W.-X. Zhou, "Matrix completion via max-norm constrained optimization," *Electron. J. Stat.*, vol. 10, no. 1, pp. 1493–1525, 2016.

[7] J. Brodie, I. Daubechiesa, C. D. Mol, D. Giannone, and I. Loris, "Sparse and stable markowitz portfolios," *Proc. Natl. Acad. Sci.*, vol. 106, no. 30, pp. 12267–12272, 2009.

[8] A. Yurtsever, B. C. Vu, and V. Cevher, "Stochastic three-composite convex minimization," in *Proc. NIPS*, pp. 4329–4337, 2016.

[9] D. Davis and W. Yin, "A three-operator splitting scheme and its optimization applications," *Set-Valued Var. Anal.*, 2017.

[10] H. Ouyang and A. G. Gray, "Stochastic smoothing for nonsmooth minimizations: Accelerating SGD by exploiting structure," in *Proc. ICML*, (Edinburgh, UK), 2012.

[11] W. Zhong and J. Kwok, "Accelerated stochastic gradient method for composite regularization," in *Proc. AISTATS*, (Reykjavik, Iceland), pp. 1086–1094, 2014.

[12] Q. Lin, X. Chen, and J. Pena, "A smoothing stochastic gradient method for composite optimization," *Optim. Methods Softw.*, vol. 29, no. 6, pp. 1281–1301, 2014.

[13] C. Hu, W. Pan, and J. T. Kwok, "Accelerated gradient methods for stochastic optimization and online learning," in *Porc. NIPS*, pp. 781–789, 2009.

[14] L. Rosasco, S. Villa, and B. C. V, "Convergence of stochastic proximal gradient algorithm." arXiv:1403.5074, 2014.

[15] Y. F. Atchadé, G. Fort, and E. Moulines, "On perturbed proximal gradient algorithms," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 310–342, 2017.

[16] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1574–1609, 2009.

[17] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *J. Mach. Learn. Res.*, vol. 10, pp. 2899–2934, 2009.

[18] G. Lan, "An optimal method for stochastic composite optimization," *Math. Program.*, vol. 133, no. 1-2, pp. 365–397, 2012.

[19] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework," *SIAM J. Optim.*, vol. 22, no. 4, pp. 1469–1492, 2012.

[20] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: Shrinking procedures and optimal algorithms," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2061–2089, 2013.

[21] O. Shamir and T. Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in *Proc. ICML*, pp. 71–79, 2013.

[22] E. Hazan and S. Kale, "Beyond the regret minimization barrier: Optimal algorithms for stochastic strongly-convex optimization," *J. Mach. Learn. Res.*, vol. 15, pp. 2489–2512, 2014.

[23] Y. Nesterov, "Smooth minimization of nonsmooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, 2005.

[24] R. T. Rockafellar, *Convex analysis*. Princeton University Press, 1970.

[25] H. H. Bauschke, R. Goebel, Y. Lucet, and X. Wang, "The proximal average: Basic theory," *SIAM J. Optim.*, vol. 19, no. 2, pp. 766–785, 2008.

[26] Y. Yu, "Better approximation and faster algorithm using the proximal average," in *Proc. NIPS*, pp. 458–466, 2013.

[27] T. Suzuki, "Dual averaging and proximal gradient descent for online alternating direction multiplier method," in *Proc. ICML*, pp. 392–400, 2013.

[28] S. Azadi and S. Sra, "Towards an optimal stochastic alternating direction method of multipliers," in *Proc. ICML*, (Beijing, China), pp. 620–628, 2014.

[29] P. L. Combettes and J.-C. Pesquet, "Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators," *Set-Valued Var. Anal.*, vol. 20, no. 2, pp. 307–330, 2012.

[30] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, pp. 460–479, Aug 2013.

[31] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Adv. Comput. Math.*, vol. 38, pp. 667–681, Apr 2013.

[32] R. I. Boţ, E. R. Csetnek, A. Heinrich, and C. Hendrich, "On the convergence rate improvement of a primal-dual splitting algorithm for solving monotone inclusion problems," *Math. Program.*, vol. 150, no. 2, pp. 251–279, 2015.

[33] N. He, A. Juditsky, and A. Nemirovski, "Mirror prox algorithm for multi-term composite minimization and semi-separable problems," *Comput. Optim. Appl.*, vol. 61, no. 2, pp. 275–319, 2015.

[34] A. Chambolle and T. Pock, "On the ergodic convergence rates of a first-order primal–dual algorithm," *Math. Program.*, vol. 159, no. 1, pp. 253–287, 2016.

[35] Y. He and R. D. C. Monteiro, "An accelerated hpe-type algorithm for a class of composite convex-concave saddle-point problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 29–56, 2016.

[36] Q. V. Nguyen, O. Fercoq, and V. Cevher, "Smoothing technique for nonsmooth composite minimization with linear operator." arXiv:1706.05837, 2017.

[37] A. Alacaoglu, Q. T. Dinh, O. Fercoq, and V. Cevher, "Smooth primal-dual coordinate descent algorithms for nonsmooth convex optimization," in *Proc. NIPS*, 2017.

[38] Y. Chen, G. Lan, and Y. Ouyang, "Optimal primal-dual methods for a class of saddle point problems," *SIAM J. Optim.*, vol. 24, no. 4, pp. 1779–1814, 2014.

[39] A. Juditsky, A. Nemirovski, and C. Tauvel, "Solving variational inequalities with stochastic mirror-prox algorithm," *Stoch. Syst.*, vol. 1, no. 1, pp. 17–58, 2011.

[40] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, 2011.

[41] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 1951.

[42] H. J. Kushner and G. Yin, *Stochastic approximation and recursive algorithms and applications*. Springer, 2003.

[43] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 1964.

[44] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," in *Proc. ICML*, (Edinburgh, Scotland), pp. 1571–1578, 2012.

[45] L. Qiao, T. Lin, Y. Jiang, F. Yang, W. Liu, and X. Lu, "On stochastic primal-dual hybrid gradient approach for compositely regularized minimization," in *Proc. ECAI*, (Hague, Netherlands), pp. 167–174, 2016.

[46] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in *Proc. ICML*, (Montreal, Quebec, Canada), pp. 433–440, 2009.

[47] E. X. Fang, H. Liu, K.-C. Toh, and W.-X. Zhou, "Max-norm optimization for robust matrix recovery," *Math. Program.*, 2017.

[48] S. R. Becker and P. L. Combettes, "An algorithm for splitting parallel sums of linearly composed monotone operators, with applications to signal recovery," *J. Nonlinear and Convex Anal.*, vol. 15, no. 1, pp. 137–159, 2014.

[49] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing, "Smoothing proximal gradient method for general structured sparse regression," *Ann. Appl. Stat.*, vol. 6, no. 2, pp. 719–752, 2012.

[50] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.