Accelerated Dual-Averaging Primal-Dual Method for Composite Convex Minimization

Conghui Tan[a], Yuqiu Qian[b], Shiqian Ma[c] and Tong Zhang[d]

[a]The Chinese University of Hong Kong; [b]University of Hong Kong;
[c]University of California, Davis; [d]The Hong Kong University of Science and Technology

ARTICLE HISTORY
Compiled October 18, 2019

ABSTRACT
Dual averaging-type methods are widely used in industrial machine learning applications due to their ability to promoting solution structure (e.g., sparsity) efficiently. In this paper, we propose a novel accelerated dual-averaging primal-dual algorithm for minimizing a composite convex function. We also derive a stochastic version of the proposed method which solves empirical risk minimization, and its advantages on handling sparse data are demonstrated both theoretically and empirically.

KEYWORDS
Dual Averaging Algorithm; Primal-dual; Empirical Risk Minimization; Acceleration; Sparse Data

1. Introduction

In this paper, we consider minimizing the following composite convex function:

$$\min_{x\in\mathbb{R}^d} \{P(x) \coloneqq f(Ax) + g(x)\}, \tag{1}$$

where $A \in \mathbb{R}^{n\times d}$, and both $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ are convex closed functions. Here f can be either smooth or non-smooth, and we assume g has easy proximal mapping. Problem (1) covers a wide range of applications. For example, choosing f to be the indicator function of a convex set $C = \{z \in \mathbb{R}^n | z \le b\}$ corresponds to minimizing a convex function over a polyhedron. It covers the Lasso problem [21]

$$\min_{x\in\mathbb{R}^d} \left\{ \frac{1}{2n}\|Ax - b\|_2^2 + \lambda\|x\|_1 \right\}, \tag{2}$$

Dedicated to Professor Ya-xiang Yuan on the occasion of his 60th birthday.

Corresponding Author. Email: sqma@ucdavis.edu

by setting $f(u) = \frac{1}{2n}\|u-b\|_2^2$ and $g(x) = \lambda\|x\|_1$. Another application of the form (1) is the support vector machine (SVM):

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max\left\{1 - \langle b_i a_i, x\rangle, 0\right\} + \frac{\lambda}{2}\|x\|_2^2, \tag{3}$$

where $a_i \in \mathbb{R}^d$ is the feature vector of the i-th data sample, and $b_i \in \{\pm 1\}$ is the corresponding label.

For smooth f, a classical way to solve (1) is the proximal gradient method (PGM) and its accelerations [2, 22]. PGM for solving (1) iterates as

$$x^{t+1} = \mathrm{prox}_{\eta g}\left(x^t - \eta A^\top \nabla f(Ax^t)\right),$$

where $\eta > 0$ is the step size. Dual averaging (DA, [16]) algorithm is another widely used algorithm for solving (1), which iterates as

$$x^{t+1} = \mathrm{prox}_{\sum_{k=0}^{t}\beta_t g}\left(x^0 - \sum_{k=0}^{t}\beta_k A^\top \nabla f(Ax^k)\right),$$

where $\{\beta_t\}$ are the step sizes. Different from PGM, in each iteration, DA always starts at the initial iterate x^0, averages all the past gradients, and then conducts proximal mapping. Dual-averaging type methods are widely used in many industrial machine learning applications due to the following advantages over PGM [5, 13, 14]. First, it is observed that DA is better in promoting solution structure (e.g., sparsity) than PGM [12, 23]. Second, DA can deal with sparse data much more efficiently than PGM. We will provide more details in Section 4.

In this paper, we develop a new dual-averaging primal-dual (DAPD) method for solving (1), which has accelerated optimal convergence rate. When $f(Ax)$ has a finite-sum structure, we develop a stochastic version of DAPD, named SDAPD, which is also optimal, and has better overall complexity on sparse data comparing with existing algorithms of the same type.

Notation. The following notation is adopted throughout this paper. For the matrix $A \in \mathbb{R}^{n\times d}$ used in (1), we use $a_i^\top$ to denote the i-th row of A and a_{ij} to denote the j-th coordinate of a_i ($1 \le i \le n$, $1 \le j \le d$). We define

$$R \coloneqq \|A\|_2 \quad \text{and} \quad \bar{R} \coloneqq \max_{i=1,\dots,n}\|a_i\|_2. \tag{4}$$

Note that $\|z\|_2$ denotes the spectral norm if z is a matrix, and ℓ_2 norm if z is a vector. It is easy to show that R and $\bar{R}$ have the following relationship: $\bar{R} \le R \le \sqrt{n}\bar{R}$. We use ρ to denote the proportion of non-zero entries in A (note $0 < \rho \le 1$). To ease the later discussion on computational complexity, without loss of generality, we assume $\rho \ge 1/n$ and $\rho \ge 1/d$, which happens for large-scale problems. For a convex set C, $\mathrm{dist}\,(x, C) \coloneqq \inf_{x'\in C}\|x - x'\|_2$ is the distance between point x and set C. For any function $h(u) : \mathbb{R}^p \to \mathbb{R}$, its proximal mapping is defined as:

$$\mathrm{prox}_h(u) \coloneqq \arg\min_{v\in\mathbb{R}^p}\left\{h(v) + \frac{1}{2}\|v-u\|_2^2\right\}, \quad \forall u \in \mathbb{R}^p.$$

Algorithm 1 Dual-Averaging Primal-Dual (DAPD) Method

Input: initial points x^0 and y^0, primal and dual step sizes $\{\beta_t\}$, $\{\eta_t\}$ and $\{\tau_t\}$

1: Initialize $B_0 = \beta_0$

2: **for** $t = 0, 1, 2, \ldots$ **do**

3: Compute intermediate variable:

$$\bar{x}^{t+1} := \mathrm{prox}_{\eta_t g}\left(x^t - \eta_t A^\top y^t\right) \tag{6}$$

4: Update dual variable:

$$y^{t+1} := \mathrm{prox}_{\tau_t f^*}\left(y^t + \tau_t A \bar{x}^{t+1}\right) \tag{7}$$

5: Update primal variable via a dual-averaged step:

$$x^{t+1} := \mathrm{prox}_{B_t g}\left(x^0 - \sum_{k=0}^{t} \beta_k A^\top y^{k+1}\right) \tag{8}$$

6: Update $B_{t+1} := B_t + \beta_{t+1}$

7: **end for**

The domain of function $h(u)$ is denoted as $\mathrm{dom}\, h := \{u \in \mathbb{R}^p | h(u) < +\infty\}$ and its conjugate function is defined as $h^*(v) = \sup_{u \in \mathbb{R}^p}\{\langle v, u\rangle - h(u)\}$. $\partial h(u)$ denotes the subdifferential of h at u. The function $h(u)$ is said to be μ-strongly convex if

$$h(v) \geq h(u) + \langle s, v - u\rangle + \frac{\mu}{2}\|v - u\|_2^2, \quad \forall s \in \partial h(u),\ u, v \in \mathbb{R}^p.$$

$h(u)$ is called L-Lipschitz continuous if it satisfies

$$|h(u) - h(v)| \leq L\|u - v\|_2, \quad \forall u, v \in \mathbb{R}^p.$$

$h(u)$ is called ζ-smooth if it is differentiable and its gradient is ζ-Lipschitz continuous, i.e.,

$$\|\nabla h(u) - \nabla h(v)\|_2 \leq \zeta\|u - v\|_2, \quad \forall u, v \in \mathbb{R}^p.$$

2. The Dual-Averaging Primal-Dual Algorithm

In this section, we present our dual-averaging primal-dual (DAPD) algorithm, which solves the following primal-dual formulation of problem (1):

$$\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} \left\{F(x, y) := g(x) + \langle y, Ax\rangle - f^*(y)\right\}. \tag{5}$$

We use (x^*, y^*) to denote a pair of optimal primal-dual solutions to (5), and X^* and Y^* the sets containing all optimal primal and dual solutions, respectively.

The details of DAPD algorithm are presented in Algorithm 1. In each iteration, DAPD first conducts one primal proximal gradient step to compute the intermediate

variable $\bar{x}^{t+1}$, and then y^{t+1} is computed using the gradient evaluated at $\bar{x}^{t+1}$. Finally, x^{t+1} is updated in (8), which adopts a dual-averaging type update rule. Note that all the past dual intermediate variables $\{y^{k+1}\}_{k=0}^{t}$ play a role here, and the gradient used in (8) is a weighted sum of them, instead of simply y^{t+1}. The update of y^{t+1} in (7) can be viewed as an extragradient step [8, 9], since the gradient used here is evaluated at the intermediate variable $\bar{x}^{t+1}$ instead of x^t. Moreover, (7) and (8) have the flavor of the primal-dual hybrid gradient [3]. Note that the step size used in the proximal mapping in (8) is B_t, which is much larger than β_t. This helps promote the desired structures of solution x^t. For instance, if g is the ℓ_1 norm, then x^{t+1} generated by (8) is more likely to be sparse because B_t is large.

When implementing DAPD, the summation in (8) needs to be computed incrementally. By doing so, the main computation cost in each iteration of Algorithm 1 lies in the matrix-vector multiplications $A^\top y^t$, $A\bar{x}^{t+1}$ and $A^\top y^{t+1}$. Since A is a n-by-d matrix with sparsity ρ, these multiplications can be done in $\mathcal{O}(\rho nd)$ operations.

We now analyze the convergence rate of DAPD (Algorithm 1). The following assumption is made throughout this section.

Assumption 2.1. *f is $(1/\gamma)$-smooth ($\gamma \geq 0$), and $g(x)$ is μ-strongly convex ($\mu \geq 0$).*

Note that $\gamma = 0$ means that f is non-smooth, and $\mu = 0$ means that g is non-strongly convex.

Although some parts of our DAPD algorithm look very similar to the primal-dual hybrid gradient (PDHG, [3]), technical challenges still exist if we want to directly adapt the analysis of PDHG to our algorithm.

(i) $\bar{x}^{t+1}$ in DAPD is obtained by a gradient step instead of extrapolation step. In the analysis of PDHG, the extrapolation step plays an important role in canceling the mismatch between primal and dual variables. Here we need a new approach to tackle this difficulty.

(ii) Since the primal updates consist of two gradient descent steps, two very different sequences of primal step sizes $\{\eta_t\}$ and $\{\beta_t\}$ and the dual step size $\{\tau_t\}$ need to be specified. This requires us to carefully balance these three parameters so that we can obtain the fastest convergence.

(iii) The update of x^{t+1} in DAPD is in the dual averaging style, which is very different from PDHG in that it involves all the past gradients rather than simply the gradient at y^{t+1}. This makes it difficult to relate this step to the objective function value $F(x^{t+1}, y^{t+1})$.

In order to tackle these issues, new techniques are needed for the analysis. We define a potential function ϕ_t to characterize the dual-averaging step as follows:

$$\phi_t(x) := \frac{1}{2}\left\|x - x^0\right\|_2^2 + \sum_{k=0}^{t-1} \beta_k \left(g(x) + \langle y^{k+1}, Ax\rangle\right). \tag{9}$$

From (8) it is easy to observe that $x^{t+1} := \arg\min_x \phi_{t+1}(x)$. Besides, since $g(x)$ is μ-strongly convex, $\phi_t(x)$ is strongly convex with strong convexity parameter $1 + \sum_{k=0}^{t-1} \beta_k \mu = 1 + B_{t-1}\mu$. Moreover, we denote $\phi_t^* := \min_{x\in\mathbb{R}^d} \phi_t(x)$.

The following lemma characterizes the change of ϕ_t^* after one iteration.

Lemma 2.2. *Assume*

$$\eta_t(1 + B_{t-1}\mu) \geq \beta_t. \tag{10}$$

We have

$$\phi_{t+1}^* - \phi_t^* \geq \beta_t \left(g(\bar{x}^{t+1}) + \langle y^{t+1}, A\bar{x}^{t+1}\rangle\right) - \frac{\beta_t R^2 \eta_t}{2}\|y^{t+1} - y^t\|_2^2. \tag{11}$$

Proof. From the strong convexity of $\phi_{t+1}(x)$ and (10), we obtain

$$\begin{aligned}
\phi_{t+1}^* =& \phi_{t+1}(x^{t+1}) = \phi_t(x^{t+1}) + \beta_t \left(g(x^{t+1}) + \langle y^{t+1}, Ax^{t+1}\rangle\right) \\
\geq& \phi_t^* + \frac{1 + B_{t-1}\mu}{2}\|x^t - x^{t+1}\|_2^2 + \beta_t \left(g(x^{t+1}) + \langle y^{t+1}, Ax^{t+1}\rangle\right) \\
\geq& \phi_t^* + \frac{\beta_t}{2\eta_t}\|x^t - x^{t+1}\|_2^2 + \beta_t \left(g(x^{t+1}) + \langle y^{t+1}, Ax^{t+1}\rangle\right).
\end{aligned} \tag{12}$$

Note that (6) can be rewritten as $\bar{x}^{t+1} = x^t - \eta_t \left(A^\top y^t + s\right),\ \exists s \in \partial g(\bar{x}^{t+1})$, which yields

$$\begin{aligned}
&\|x^t - x^{t+1}\|_2^2 - \|x^t - \bar{x}^{t+1}\|_2^2 - \|\bar{x}^{t+1} - x^{t+1}\|_2^2 \\
=& 2\langle x^t - \bar{x}^{t+1}, \bar{x}^{t+1} - x^{t+1}\rangle = 2\eta_t\langle A^\top y^t + s, \bar{x}^{t+1} - x^{t+1}\rangle \\
\geq& 2\eta_t \left(\langle y^t, A(\bar{x}^{t+1} - x^{t+1})\rangle + g(\bar{x}^{t+1}) - g(x^{t+1})\right),
\end{aligned} \tag{13}$$

where the inequality is due to the convexity of $g(x)$. Combining (12) and (13) yields

$$\begin{aligned}
&\phi_{t+1}^* \\
\geq& \phi_t^* + \frac{\beta_t}{2\eta_t}\left[\|x^t - \bar{x}^{t+1}\|_2^2 + \|\bar{x}^{t+1} - x^{t+1}\|_2^2 + 2\eta_t \left(\langle y^t, A(\bar{x}^{t+1} - x^{t+1})\rangle + g(\bar{x}^{t+1}) - g(x^{t+1})\right)\right] \\
&+ \beta_t \left(g(x^{t+1}) + \langle y^{t+1}, Ax^{t+1}\rangle\right) \\
=& \phi_t^* + \frac{\beta_t}{2\eta_t}\left(\|x^t - \bar{x}^{t+1}\|_2^2 + \|\bar{x}^{t+1} - x^{t+1}\|_2^2\right) + \beta_t \left(g(\bar{x}^{t+1}) + \langle y^{t+1}, A\bar{x}^{t+1}\rangle\right) \\
&+ \beta_t\langle y^t - y^{t+1}, A(\bar{x}^{t+1} - x^{t+1})\rangle \\
\geq& \phi_t^* + \frac{\beta_t}{2\eta_t}\left(\|x^t - \bar{x}^{t+1}\|_2^2 + \|\bar{x}^{t+1} - x^{t+1}\|_2^2\right) + \beta_t \left(g(\bar{x}^{t+1}) + \langle y^{t+1}, A\bar{x}^{t+1}\rangle\right) \\
&- \beta_t \left(\frac{R^2\eta_t}{2}\|y^{t+1} - y^t\|_2^2 + \frac{1}{2R^2\eta_t}\|A(\bar{x}^{t+1} - x^{t+1})\|_2^2\right) \\
\geq& \phi_t^* + \frac{\beta_t}{2\eta_t}\left(\|x^t - \bar{x}^{t+1}\|_2^2 + \|\bar{x}^{t+1} - x^{t+1}\|_2^2\right) + \beta_t \left(g(\bar{x}^{t+1}) + \langle y^{t+1}, A\bar{x}^{t+1}\rangle\right) \\
&- \beta_t \left(\frac{R^2\eta_t}{2}\|y^{t+1} - y^t\|_2^2 + \frac{1}{2\eta_t}\|\bar{x}^{t+1} - x^{t+1}\|_2^2\right) \\
\geq& \phi_t^* - \frac{\beta_t R^2 \eta_t}{2}\|y^{t+1} - y^t\|_2^2 + \beta_t \left(g(\bar{x}^{t+1}) + \langle y^{t+1}, A\bar{x}^{t+1}\rangle\right),
\end{aligned}$$

where the second inequality is due to Young's inequality and the third inequality is from (4). This completes the proof. □

The next lemma concerns the update of the dual variable.

Lemma 2.3. *For any $y \in \mathbb{R}^n$, it holds that*

$$\begin{aligned}&\frac{1}{2\tau_t}\left(\|y^t - y\|_2^2 - (1+\gamma\tau_t)\|y^{t+1} - y\|_2^2 - \|y^{t+1} - y^t\|_2^2\right)\\ \geq&\langle A\bar{x}^{t+1}, y - y^{t+1}\rangle + f^*(y^{t+1}) - f^*(y).\end{aligned} \tag{14}$$

Proof. Using (7) and following similar derivation as (13), it is easy to show that there exists $s \in \partial f^*(y^{t+1})$ such that the following holds:

$$\begin{aligned}&\|y^t - y\|_2^2 - \|y^{t+1} - y\|_2^2 - \|y^t - y^{t+1}\|_2^2\\ =&2\langle y^t - y^{t+1}, y^{t+1} - y\rangle = 2\langle \tau_t(-A\bar{x}^{t+1} + s), y^{t+1} - y\rangle\\ \geq&2\tau_t\left(\langle A\bar{x}^{t+1}, y - y^{t+1}\rangle + f^*(y^{t+1}) - f^*(y) + \frac{\gamma}{2}\|y^{t+1} - y\|_2^2\right),\end{aligned} \tag{15}$$

where the inequality is due to the γ-strong convexity of $f^*(y)$, which is implied by the $(1/\gamma)$-smoothness of f [7]. Dividing (15) by $2\tau_t$ yields (14). □

We are now ready to present the main convergence results of DAPD.

Theorem 2.4. *Consider the first T iterations of DAPD. Assume the parameters satisfy* (10) *and the following conditions:*

$$\eta_t\tau_t \leq \frac{1}{R^2}, \tag{16}$$

$$\frac{\beta_{t+1}}{\tau_{t+1}} \leq \frac{\beta_t}{\tau_t}(1+\gamma\tau_t). \tag{17}$$

Define

$$\hat{x}^T = \frac{1}{B_{t-1}}\sum_{t=0}^{T-1}\beta_t\bar{x}^{t+1} \quad \textit{and} \quad \hat{y}^T = \frac{1}{B_{t-1}}\sum_{t=0}^{T-1}\beta_t y^{t+1}. \tag{18}$$

The following inequality holds for any $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^n$:

$$F(\hat{x}^T, y) - F(x, \hat{y}^T) \leq \frac{1}{B_{T-1}}\left(\frac{\beta_0}{2\tau_0}\|y^0 - y\|_2^2 + \frac{1}{2}\|x^0 - x\|_2^2\right). \tag{19}$$

Proof. Multiplying (14) by β_t, and adding the resulted inequality to (11), we obtain

$$\begin{aligned}&\phi_{t+1}^* - \phi_t^* + \frac{\beta_t}{2\tau_t}\left(\|y^t - y\|_2^2 - (1+\gamma\tau_t)\|y^{t+1} - y\|_2^2\right)\\ \geq&\beta_t\left(g(\bar{x}^{t+1}) + \langle y^{t+1}, A\bar{x}^{t+1}\rangle\right) - \frac{\beta_t R^2\eta_t}{2}\|y^{t+1} - y^t\|_2^2\\ &+ \beta_t\left(\langle y - y^{t+1}, A\bar{x}^{t+1}\rangle + f^*(y^{t+1}) - f^*(y)\right) + \frac{\beta_t}{2\tau_t}\|y^{t+1} - y^t\|_2^2\\ \geq&\beta_t\left(\langle y, A\bar{x}^{t+1}\rangle + g(\bar{x}^{t+1}) + f^*(y^{t+1}) - f^*(y)\right),\end{aligned} \tag{20}$$

where the last inequality is due to (16). Combining (17) and (20) yields

$$\begin{aligned}&\left(\frac{\beta_t}{2\tau_t}\|y^t-y\|_2^2-\phi_t^*\right)-\left(\frac{\beta_{t+1}}{2\tau_{t+1}}\|y^{t+1}-y\|_2^2-\phi_{t+1}^*\right)\\ \geq&\beta_t\left(\langle y,A\bar{x}^{t+1}\rangle+g(\bar{x}^{t+1})+f^*(y^{t+1})-f^*(y)\right).\end{aligned}\tag{21}$$

Note that the left hand side of (21) has a telescoping structure. Summing (21) over $t=0,\ldots,T-1$ yields

$$\begin{aligned}&\sum_{t=0}^{T-1}\beta_t\left(\langle y,A\bar{x}^{t+1}\rangle+g(\bar{x}^{t+1})+f^*(y^{t+1})-f^*(y)\right)\\ \leq&\left(\frac{\beta_0}{2\tau_0}\|y^0-y\|_2^2-\phi_0^*\right)-\left(\frac{\beta_T}{2\tau_T}\|y^T-y\|_2^2-\phi_T^*\right)\leq\frac{\beta_0}{2\tau_0}\|y^0-y\|_2^2-\phi_0^*+\phi_T^*.\end{aligned}\tag{22}$$

From (9), it is straightforward that $\phi_0^*=\min_{x\in\mathbb{R}^d}\frac{1}{2}\|x-x^0\|_2^2=0$ and

$$\phi_T^*\leq\phi_T(x)=\frac{1}{2}\|x-x^0\|_2^2+\sum_{t=0}^{T-1}\beta_t\left(g(x)+\langle y^{t+1},Ax\rangle\right).$$

Combining these facts with (22) and using the convexity-concavity of $F(x,y)$, we have

$$\begin{aligned}&\frac{\beta_0}{2\tau_0}\|y^0-y\|_2^2+\frac{1}{2}\|x^0-x\|_2^2\\ \geq&\sum_{t=0}^{T-1}\beta_t\left(\langle y,A\bar{x}^{t+1}\rangle-f^*(y)+g(\bar{x}^{t+1})-\langle y^{t+1},Ax\rangle+f^*(y^{t+1})-g(x)\right)\\ =&\sum_{t=0}^{T-1}\beta_t\left(F(\bar{x}^{t+1},y)-F(x,y^{t+1})\right)\geq\left(\sum_{t=0}^{T-1}\beta_t\right)\cdot\left(F(\hat{x}^T,y)-F(x,\hat{y}^T)\right)\\ =&B_{T-1}\left(F(\hat{x}^T,y)-F(x,\hat{y}^T)\right),\end{aligned}$$

which completes the proof. □

From Theorem 2.4, we can derive some more interpretable complexity bounds by choosing some specific parameters.

Corollary 2.5. *The following facts hold for DAPD (Algorithm 1).*

(i) If $\gamma>0$ and $\mu>0$, by choosing

$$\eta_t=\frac{1}{R}\sqrt{\frac{\gamma}{\mu}},\ \ \tau_t=\frac{1}{R}\sqrt{\frac{\mu}{\gamma}}\ \ \textit{and}\ \ \beta_t=\frac{1}{R}\sqrt{\frac{\gamma}{\mu}}\left(1+\frac{\sqrt{\mu\gamma}}{R}\right)^t,\tag{23}$$

DAPD converges linearly:

$$\|\hat{x}^T-x^*\|_2^2\leq\frac{1}{\left(1+\frac{\sqrt{\mu\gamma}}{R}\right)^T-1}\left[\|x^0-x^*\|_2^2+\frac{\gamma}{\mu}\|y^0-y^*\|_2^2\right].\tag{24}$$

(ii) If $\gamma > 0$, $\mu = 0$ and f is L-Lipschitz continuous, by choosing

$$\eta_t = \beta_t = \frac{\gamma(t+1)}{3R^2} \ \text{ and } \ \tau_t = \frac{3}{\gamma(t+1)},$$

DAPD converges sublinearly in terms of primal sub-optimality:

$$P(\hat{x}^T) - P(x^*) \leq \frac{9R^2 \mathrm{dist}^2\left(x^0, X^*\right) + 4\gamma^2 L^2}{3\gamma T(T+1)}. \tag{25}$$

(iii) If $\mu > 0$ and $\gamma = 0$, by choosing

$$\eta_t = \frac{4}{\mu(t+1)}, \ \ \tau_t = \frac{\mu(t+1)}{4R^2} \ \text{ and } \ \beta_t = \frac{2(t+1)}{\mu},$$

DAPD converges sublinearly:

$$\|\hat{x}^T - x^*\|_2^2 \leq \frac{\mu\|x^0 - x^*\|_2^2 + 8R^2 \mathrm{dist}^2\left(y^0, Y^*\right)}{\mu T(T+1)}.$$

(iv) If $\gamma = 0$, $\mu = 0$ and f is L-Lipschitz continuous, by setting

$$\tau_t \equiv \tau \ \text{ and } \ \eta_t = \beta_t \equiv \frac{1}{\tau R^2},$$

where $\tau > 0$ is an arbitrary constant, we have

$$P(\hat{x}^T) - P(x^*) \leq \frac{\tau R^2 \cdot \mathrm{dist}^2\left(x^0, X^*\right) + \frac{4L^2}{\tau}}{2T}.$$

Proof. For the sake of succinctness, we only prove the first two cases, while the other two cases can be proved similarly.

Case (i): $\gamma > 0$ and $\mu > 0$. It is easy to verify that the parameter setting in (23) satisfies (10), (16) and (17). Thus, Theorem 2.4 applies here. Choosing $(x, y) = (x^*, y^*)$ in (19) gives

$$F(\hat{x}^T, y^*) - F(x^*, \hat{y}^T) \leq \frac{1}{B_{T-1}} \left(\frac{\beta_0}{2\tau_0}\|y^0 - y^*\|_2^2 + \frac{1}{2}\|x^0 - x^*\|_2^2 \right). \tag{26}$$

The μ-strong convexity of $F(\cdot, y^*)$ implies

$$F(\hat{x}^T, y^*) - F(x^*, \hat{y}^T) \geq F(\hat{x}^T, y^*) - F(x^*, y^*) \geq \frac{\mu}{2}\|\hat{x}^T - x^*\|_2^2. \tag{27}$$

Combining (26), (27) and (23) yields (24).

Case (ii): $\gamma > 0$, $\mu = 0$ and f is Lipschitz continuous. It is again easy to verify that the conditions in Theorem 2.4 are satisfied and thus Theorem 2.4 applies here. In (19), we set $x = x^*$ and take supremum with respect to y in the domain of f^*, which

gives

$$\frac{1}{B_{T-1}}\left(\frac{\beta_0}{2\tau_0}\sup_{y\in\operatorname{dom} f^*}\|y^0-y\|_2^2+\frac{1}{2}\|x^0-x^*\|_2^2\right)\geq\sup_{y\in\operatorname{dom} f^*}F(\hat{x}^T,y)-F(x^*,\hat{y}^T)$$
$$\geq P(\hat{x}^T)-P(x^*). \tag{28}$$

Because f is L-Lipschitz continuous, the domain of f^* is bounded such that $\|y\|_2\leq L$ for all $y\in\operatorname{dom} f^*$ [20]. Hence, (28) implies

$$\frac{1}{B_{T-1}}\left(\frac{2\beta_0}{\tau_0}L^2+\frac{1}{2}\|x^0-x^*\|_2^2\right)\geq P(\hat{x}^T)-P(x^*). \tag{29}$$

Since (29) holds for any $x^*\in X^*$, by replacing $\|x^0-x^*\|_2^2$ by $\operatorname{dist}^2\left(x^0,X^*\right)$ in (29) we obtain the desired result (25). □

Remark 1. For problem (1), if f is $(1/\gamma)$-smooth and g is μ-strongly convex, the condition number of problem (1) is $\kappa:=\frac{R^2}{\mu\gamma}$. The case (i) in Corollary 2.5 implies that DAPD requires $\mathcal{O}\left(\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$ iterations to achieve ϵ accuracy, which is an accelerated rate and matches the complexity lower bound of first-order methods.

On the other hand, when the objective function of (1) is smooth but non-strongly convex (case (ii)), or is non-smooth but strongly convex (case (iii)), Corollary 2.5 implies that DAPD has $\mathcal{O}\left(\frac{1}{T^2}\right)$ accelerated convergence rate, which is also optimal for first-order methods. For non-smooth and non-strongly convex problems (case (iv)), the convergence rate of DAPD is $\mathcal{O}\left(\frac{1}{T}\right)$, which is faster than subgradient method and the original dual averaging method, whose rates are $\mathcal{O}(1/\sqrt{T})$ under the same assumptions.

The assumption that f is Lipschitz continuous required in cases (ii) and (iv) of Corollary 2.5 is standard for primal-dual methods.

Remark 2. Though our theoretical analysis is based on the averaged iterates $(\hat{x}^T,\hat{y}^T)$, in the actual implementation of our algorithms, we will always choose the last iterate (x^T,y^T) as the output to make sure the solution structure (e.g., sparsity) will be preserved. Such strategy is also the common practice of dual-averaging-type methods [23].

3. The Stochastic DAPD Method

In this section, we focus on (1) where f has a finite-sum structure. More specifically, we assume that the primal problem is of the following form:

$$\min_{x\in\mathbb{R}^d}\left\{\tilde{P}(x):=\frac{1}{n}\sum_{i=1}^n f_i(a_i^\top x)+g(x)\right\}, \tag{30}$$

Algorithm 2 Stochastic Dual-Averaging Primal-Dual (SDAPD) Method

Input: initial values x^0 and y^0, primal step sizes $\{\beta_t\}$ and η, dual step size τ

1: Initialize $\bar{x}^0 = x^0$ and $B_0 = \beta_0$

2: **for** $t = 0, 1, \ldots$ **do**

3: Uniformly randomly sample $i_t \in \{1, 2, \ldots, n\}$

4: Compute intermediate variable:

$$\bar{x}^{t+1} = \mathrm{prox}_{\eta g}\left(x^t - \frac{\eta}{n} A^\top y^t\right) \tag{31}$$

5: Update dual variable:

$$y_i^{t+1} = \begin{cases} \tilde{y}_i^{t+1} \coloneqq \mathrm{prox}_{\tau f_i^*}\left(y_i^t + \tau \langle a_i, \bar{x}^{t+1}\rangle\right), & \text{if } i = i_t \\ y_i^t, & \text{if } i \neq i_t \end{cases} \tag{32}$$

6: Set

$$\bar{y}^{t+1} = y^t + n(y^{t+1} - y^t) \tag{33}$$

7: Update primal variable:

$$x^{t+1} = \mathrm{prox}_{B_t g}\left(x^0 - s^{t+1}\right), \text{ with } s^{t+1} \coloneqq \sum_{k=0}^{t} \frac{\beta_k}{n} A^\top \bar{y}^{k+1} \tag{34}$$

8: Let $B_{t+1} \coloneqq B_t + \beta_{t+1}$

9: **end for**

with $f_i : \mathbb{R} \to \mathbb{R}$. Problem (30) reduces to (1) by choosing $f(u) = \frac{1}{n}\sum_{i=1}^n f_i(u)$. The primal-dual formulation of (30) is:

$$\min_{x\in\mathbb{R}^d} \max_{y\in\mathbb{R}^n} \left\{ \tilde{F}(x, y) \coloneqq \frac{1}{n}\langle y, Ax\rangle + g(x) - \frac{1}{n}\sum_{i=1}^n f_i^*(y_i) \right\}.$$

Since (30) is a special case of (1), DAPD can be directly applied here. If we assume each f_i is $(1/\gamma)$-smooth and g is μ-strongly convex, the complexity of DAPD for solving (30) is $\mathcal{O}\left(\sqrt{\kappa'}\log\frac{1}{\epsilon}\right)$, and $\kappa' \coloneqq \frac{R^2}{n\mu\gamma}$ denotes the condition number. In this section, we show that by utilizing the finite-sum structure of f in problem (30), we can design a stochastic version of DAPD, which has a better complexity.

Our stochastic method SDAPD, which is inspired by the stochastic primal-dual coordinate (SPDC) method [27], is presented in Algorithm 2. In each iteration of SDAPD, only one coordinate of the dual variable y_{i_t} is updated, with i_t sampled uniformly random from $\{1, 2, \ldots, n\}$. Correspondingly, only one row vector $a_{i_t}^\top$ is involved in the update of the dual variable. Besides, another variable $\bar{y}^{t+1}$ is obtained by extrapolation. Moreover, note that in Algorithm 2 we only consider fixed primal and dual step sizes η and τ.

When implementing SDAPD, one should keep an auxiliary variable

$$u^t \coloneqq \frac{1}{n} A^\top y^t. \tag{35}$$

Since each time only one coordinate of y is changed, u_t can be updated incrementally as:

$$u^{t+1} = u^t + \frac{1}{n}(y_{i_t}^{t+1} - y_{i_t}^t) a_{i_t}. \tag{36}$$

As a result, the matrix-vector multiplication in (34) can be efficiently computed by:

$$\frac{1}{n} A^\top \bar{y}^{t+1} = \frac{1}{n} A^\top y^{t+1} + (y_{i_t}^{t+1} - y_{i_t}^t) a_{i_t} = u^{t+1} + (y_{i_t}^{t+1} - y_{i_t}^t) a_{i_t}.$$

Therefore, the summation of gradients s^{t+1} in (34) can also be incrementally updated with $\mathcal{O}(d)$ computation cost. As a result, the per-iteration complexity of SDAPD is $\mathcal{O}(d)$, much cheaper than the per-iteration complexity $\mathcal{O}(nd)$ of DAPD.

Remark 3. We need to point out that Murata and Suzuki also developed an accelerated stochastic dual averaging method [15] which is based on stochastic variance-reduction techniques [6] and requires the assumption that f_i is smooth.

We now provide the convergence analysis of SDAPD. Here we make the following assumption.

Assumption 3.1. *All f_i's are $(1/\gamma)$-smooth ($\gamma > 0$), and $g(x)$ is μ-strongly convex ($\mu > 0$).*

For the ease of presentation, we denote $f^*(y) \coloneqq \frac{1}{n}\sum_{i=1}^n f_i^*(y_i)$ throughout this section. Besides, we use $\mathcal{F}_t$ to stand for the σ-field generated by all random variables up to iteration t. Clearly, when conditioned on $\mathcal{F}_t$, x^t and y^t are known. Similar to the analysis of DAPD, we define a potential function as follows:

$$\tilde{\phi}_t(x) \coloneqq \frac{1}{2}\left\|x - x^0\right\|_2^2 + \sum_{k=0}^{t-1} \beta_k \left(g(x) + \frac{1}{n}\langle \bar{y}^{k+1}, Ax\rangle \right). \tag{37}$$

Again, it is easy to see that x^{t+1} is the minimizer of $\tilde{\phi}_{t+1}(x)$. Since the updates of $\bar{x}^{t+1}$ and x^{t+1} in SDAPD are almost identical to DAPD, we have the following lemma that is similar to Lemma 2.2.

Lemma 3.2. *Assume $\eta(1 + B_{t-1}\mu) \geq \beta_t$. We have*

$$\mathbb{E}\left[\tilde{\phi}_{t+1}^* - \tilde{\phi}_t^* \middle| \mathcal{F}_t\right] \geq \beta_t \mathbb{E}\left[g(\bar{x}^{t+1}) + \frac{1}{n}\langle \bar{y}^{t+1}, A\bar{x}^{t+1}\rangle \middle| \mathcal{F}_t \right] - \frac{\bar{R}^2 \beta_t \eta}{2} \mathbb{E}\left[\|y^{t+1} - y^t\|_2^2 \middle| \mathcal{F}_t \right]. \tag{38}$$

Proof. The proof is largely the same as Lemma 2.2. Following the same argument as

in Lemma 2.2, it is easy to show that (12) becomes

$$\tilde{\phi}^*_{t+1} \geq \tilde{\phi}^*_t + \frac{\beta_t}{2\eta_t}\|x^t - x^{t+1}\|_2^2 + \beta_t\left(g(x^{t+1}) + \frac{1}{n}\langle \bar{y}^{t+1}, Ax^{t+1}\rangle\right). \tag{39}$$

and (13) becomes

$$\begin{aligned}&\|x^t - x^{t+1}\|_2^2 - \|x^t - \bar{x}^{t+1}\|_2^2 - \|\bar{x}^{t+1} - x^{t+1}\|_2^2\\ \geq& 2\eta_t\left(\frac{1}{n}\langle y^t, A(\bar{x}^{t+1} - x^{t+1})\rangle + g(\bar{x}^{t+1}) - g(x^{t+1})\right).\end{aligned} \tag{40}$$

Combining (39) and (40) yields

$$\begin{aligned}&\tilde{\phi}^*_{t+1}\\ \geq&\tilde{\phi}^*_t + \frac{\beta_t}{2\eta_t}\left(\|x^t - \bar{x}^{t+1}\|_2^2 + \|\bar{x}^{t+1} - x^{t+1}\|_2^2\right) + \beta_t\left(g(\bar{x}^{t+1}) + \frac{1}{n}\langle \bar{y}^{t+1}, A\bar{x}^{t+1}\rangle\right)\\ &+ \frac{\beta_t}{n}\langle y^t - \bar{y}^{t+1}, A(\bar{x}^{t+1} - x^{t+1})\rangle\\ \geq&\tilde{\phi}^*_t + \frac{\beta_t}{2\eta_t}\left(\|x^t - \bar{x}^{t+1}\|_2^2 + \|\bar{x}^{t+1} - x^{t+1}\|_2^2\right) + \beta_t\left(g(\bar{x}^{t+1}) + \frac{1}{n}\langle \bar{y}^{t+1}, A\bar{x}^{t+1}\rangle\right)\\ &- \frac{\beta_t}{n}\left(\frac{\eta_t}{2n}\|A^\top(y^t - \bar{y}^{t+1})\|_2^2 + \frac{n}{2\eta_t}\|\bar{x}^{t+1} - x^{t+1}\|_2^2)\right).\end{aligned} \tag{41}$$

where the last inequality is due to Young's inequality. By noting that y^t and $\bar{y}^{t+1}$ only differ in coordinate i_t, we have

$$\|A^\top(y^t - \bar{y}^{t+1})\|_2^2 = \|(y^t_{i_t} - \bar{y}^{t+1}_{i_t})a_{i_t}\|_2^2 \leq (y^t_{i_t} - \bar{y}^{t+1}_{i_t})^2\bar{R}^2 = \bar{R}^2\|y^t - \bar{y}^{t+1}\|_2^2,$$

which combining with (41) yields

$$\begin{aligned}&\tilde{\phi}^*_{t+1}\\ \geq&\tilde{\phi}^*_t + \frac{\beta_t}{2\eta_t}\left(\|x^t - \bar{x}^{t+1}\|_2^2 + \|\bar{x}^{t+1} - x^{t+1}\|_2^2\right) + \beta_t\left(g(\bar{x}^{t+1}) + \frac{1}{n}\langle \bar{y}^{t+1}, A\bar{x}^{t+1}\rangle\right)\\ &- \frac{\beta_t}{n}\left(\frac{\bar{R}^2\eta_t}{2n}\|y^t - \bar{y}^{t+1}\|_2^2 + \frac{n}{2\eta_t}\|\bar{x}^{t+1} - x^{t+1}\|_2^2)\right)\\ \geq&\tilde{\phi}^*_t + \beta_t\left(g(\bar{x}^{t+1}) + \frac{1}{n}\langle \bar{y}^{t+1}, A\bar{x}^{t+1}\rangle\right) - \frac{\bar{R}^2\eta_t\beta_t}{2n^2}\|y^t - \bar{y}^{t+1}\|_2^2.\end{aligned} \tag{42}$$

Using (33) and taking conditional expectation to (42) yields the desired result (38). □

Similarly, we have the following lemma that is analogous to Lemma 2.3. We omit the proof for succinctness.

Lemma 3.3. *For each $i \in \{1, 2, \ldots, n\}$, it holds that*

$$\begin{aligned}&\frac{1}{2\tau}\left[(y_i^t - y_i)^2 - (1 + \gamma\tau)\,(\tilde{y}_i^{t+1} - y_i)^2 - (\tilde{y}_i^{t+1} - y_i^t)^2\right]\\ \geq&\langle (y_i - \tilde{y}_i^{t+1})a_i, \bar{x}^{t+1}\rangle + f_i^*(\tilde{y}_i^{t+1}) - f_i^*(y_i), \qquad \forall y_i \in \mathbb{R}.\end{aligned} \tag{43}$$

Moreover, we have the following lemma.

Lemma 3.4. *When conditioning on $\mathcal{F}_t$, for any $y \in \mathbb{R}^n$, it holds that*

$$\begin{aligned}&\frac{1}{2\tau}\mathbb{E}\left[\left(1+\frac{(n-1)\gamma\tau}{n}\right)\|y^t-y\|_2^2-(1+\gamma\tau)\|y^{t+1}-y\|_2^2-\|y^{t+1}-y^t\|_2^2\middle|\mathcal{F}_t\right]\\ \geq&\mathbb{E}\left[-\frac{1}{n}\langle\bar{y}^{t+1}-y,A\bar{x}^{t+1}\rangle+nf^*(y^{t+1})-(n-1)f^*(y^t)-f^*(y)\middle|\mathcal{F}_t\right].\end{aligned}\tag{44}$$

Proof. Note that when conditioning on $\mathcal{F}_t$, $\bar{x}^{t+1}$ is deterministic and independent of i_t. Hence, for each i, $y_i^{t+1}=\tilde{y}_i^{t+1}$ with probability $1/n$, and $y_i^{t+1}=y_i^t$ with probability $(n-1)/n$. This implies the following relationships that hold for any $y\in\mathbb{R}^n$:

$$\begin{aligned}\mathbb{E}\left[(y_i^{t+1}-y_i)^2\middle|\mathcal{F}_t\right]=&\frac{1}{n}(\tilde{y}_i^{t+1}-y_i)^2+\frac{n-1}{n}(y_i^t-y_i)^2,\\ \mathbb{E}\left[(y_i^{t+1}-y_i^t)^2\middle|\mathcal{F}_t\right]=&\frac{1}{n}(\tilde{y}_i^{t+1}-y_i^t)^2,\\ \mathbb{E}\left[y_i^{t+1}\middle|\mathcal{F}_t\right]=&\frac{1}{n}\tilde{y}_i^{t+1}+\frac{n-1}{n}y_i^t,\\ \mathbb{E}\left[f_i^*(y_i^{t+1})\middle|\mathcal{F}_t\right]=&\frac{1}{n}f_i^*(\tilde{y}_i^{t+1})+\frac{n-1}{n}f_i^*(y_i^t).\end{aligned}$$

Plugging these relationships into (43), we obtain:

$$\begin{aligned}&\frac{1}{2\tau}\mathbb{E}\left[(n+(n-1)\gamma\tau)(y_i^t-y_i)^2-n(1+\gamma\tau)(y_i^{t+1}-y_i)^2-n(y_i^{t+1}-y_i^t)^2\middle|\mathcal{F}_t\right]\\ \geq&\left\langle\left(y_i-n\mathbb{E}\left[y_i^{t+1}\middle|\mathcal{F}_t\right]+(n-1)y_i^t\right)a_i,\bar{x}^{t+1}\right\rangle+n\mathbb{E}\left[f_i^*(y_i^{t+1})\middle|\mathcal{F}_t\right]-(n-1)f_i^*(y_i^t)-f_i^*(y_i).\end{aligned}$$

Summing this inequality for $i\in\{1,2,\ldots,n\}$ and using (33), we get:

$$\begin{aligned}&\frac{1}{2\tau}\mathbb{E}\left[\left(1+\frac{(n-1)\gamma\tau}{n}\right)\|y^t-y\|_2^2-(1+\gamma\tau)\|y^{t+1}-y\|_2^2-\|y^{t+1}-y^t\|_2^2\middle|\mathcal{F}_t\right]\\ \geq&\frac{1}{n}\langle y-n\mathbb{E}\left[y^{t+1}\middle|\mathcal{F}_t\right]+(n-1)y^t,A\bar{x}^{t+1}\rangle+n\mathbb{E}\left[f^*(y^{t+1})\middle|\mathcal{F}_t\right]-(n-1)f^*(y^t)-f^*(y)\\ =&\frac{1}{n}\mathbb{E}\left[-\langle\bar{y}^{t+1}-y,A\bar{x}^{t+1}\rangle\middle|\mathcal{F}_t\right]+n\mathbb{E}\left[f^*(y^{t+1})\middle|\mathcal{F}_t\right]-(n-1)f^*(y^t)-f^*(y),\end{aligned}$$

which is the desired inequality (44). □

Now, we are ready to provide the convergence complexity for SDAPD (Algorithm 2).

Theorem 3.5. *Assume Assumption 3.1 holds. We choose algorithm parameters as*

$$\eta=\frac{1}{\bar{R}}\sqrt{\frac{\gamma}{n\mu}},\quad\tau=\frac{1}{\bar{R}}\sqrt{\frac{n\mu}{\gamma}},\quad\beta_t=\frac{1}{\bar{R}}\sqrt{\frac{\gamma}{n\mu}}\cdot\xi^t,\ \textit{with}\ \xi\coloneqq1+\frac{1}{n+\bar{R}\sqrt{n/(\mu\gamma)}}.$$

Consider the first T iterations of SDAPD and define $\hat{x}^T=\frac{1}{B_{T-1}}\sum_{t=0}^{T-1}\beta_t\bar{x}^{t+1}$, SDAPD

converges linearly in expectation:

$$\mathbb{E}\left[\|\hat{x}^T - x^*\|_2^2\right] \leq \frac{\Delta_0}{\xi^T - 1},$$

where Δ_0 is a constant depending on $\bar{R}$, the initial point (x^0, y^0) and optimal solution (x^*, y^*). Note that (x^*, y^*) is unique here due to the strong convexity-concavity assumption.

Proof. When conditioning on $\mathcal{F}_t$, we multiply (44) by β_t and add it to (38). We have

$$\begin{aligned}
&\mathbb{E}\left[\tilde{\phi}_{t+1}^* - \tilde{\phi}_t^* \middle| \mathcal{F}_t\right] + \frac{\beta_t}{2\tau}\mathbb{E}\left[\left(1 + \frac{(n-1)\gamma\tau}{n}\right)\|y^t - y\|_2^2 - (1+\gamma\tau)\|y^{t+1} - y\|_2^2 \middle| \mathcal{F}_t\right] \\
\geq &\beta_t \mathbb{E}\left[g(\bar{x}^{t+1}) + \frac{1}{n}\langle \bar{y}^{t+1}, A\bar{x}^{t+1}\rangle \middle| \mathcal{F}_t\right] - \frac{\bar{R}^2\beta_t\eta}{2}\mathbb{E}\left[\|y^{t+1} - y^t\|_2^2 \middle| \mathcal{F}_t\right] \\
&+ \beta_t \mathbb{E}\left[-\frac{1}{n}\langle \bar{y}^{t+1} - y, A\bar{x}^{t+1}\rangle + nf^*(y^{t+1}) - (n-1)f^*(y^t) - f^*(y) \middle| \mathcal{F}_t\right] \\
&+ \frac{\beta_t}{2\tau}\mathbb{E}\left[\|y^{t+1} - y^t\|_2^2 \middle| \mathcal{F}_t\right] \\
= &\beta_t \mathbb{E}\left[g(\bar{x}^{t+1}) + \frac{1}{n}\langle y, A\bar{x}^{t+1}\rangle + nf^*(y^{t+1}) - (n-1)f^*(y^t) - f^*(y) \middle| \mathcal{F}_t\right], \qquad (45)
\end{aligned}$$

where the equality uses the fact $\eta\tau = 1/\bar{R}^2$. Note that our parameters satisfy $\beta_t(1 + \gamma\tau) \geq \beta_{t+1}\alpha$, where $\alpha := 1 + \frac{(n-1)\gamma\tau}{n}$, from which we can upper bound the left-hand-side of (45) by

$$\begin{aligned}
&\mathbb{E}\left[\tilde{\phi}_{t+1}^* \middle| \mathcal{F}_t\right] - \tilde{\phi}_t^* + \frac{\alpha\beta_t}{2\tau}\|y^t - y\|_2^2 - \frac{\alpha\beta_{t+1}}{2\tau}\mathbb{E}\left[\|y^{t+1} - y\|_2^2 \middle| \mathcal{F}_t\right] \\
= &\left(\frac{\alpha\beta_t}{2\tau}\|y^t - y\|_2^2 - \tilde{\phi}_t^*\right) - \mathbb{E}\left[\frac{\alpha\beta_{t+1}}{2\tau}\|y^{t+1} - y\|_2^2 - \tilde{\phi}_{t+1}^* \middle| \mathcal{F}_t\right].
\end{aligned}$$

Therefore, (45) reduces to:

$$\begin{aligned}
&\left(\frac{\alpha\beta_t}{2\tau}\|y^t - y\|_2^2 - \tilde{\phi}_t^*\right) - \mathbb{E}\left[\frac{\alpha\beta_{t+1}}{2\tau}\|y^{t+1} - y\|_2^2 - \tilde{\phi}_{t+1}^* \middle| \mathcal{F}_t\right]. \\
\geq &\beta_t \mathbb{E}\left[g(\bar{x}^{t+1}) + \frac{1}{n}\langle y, A\bar{x}^{t+1}\rangle + nf^*(y^{t+1}) - (n-1)f^*(y^t) - f^*(y) \middle| \mathcal{F}_t\right]. \qquad (46)
\end{aligned}$$

Summing (46) over $t = 0, \ldots, T-1$ and apply total expectation, we obtain:

$$\begin{aligned}
&\left(\frac{\alpha\beta_0}{2\tau}\|y^0 - y\|_2^2 - \tilde{\phi}_0^*\right) - \mathbb{E}\left[\frac{\alpha\beta_T}{2\tau}\|y^T - y\|_2^2 - \tilde{\phi}_T^*\right] \\
\geq &\sum_{t=0}^{T-1} \beta_t \mathbb{E}\left[g(\bar{x}^{t+1}) + \frac{1}{n}\langle y, A\bar{x}^{t+1}\rangle + nf^*(y^{t+1}) - (n-1)f^*(y^t) - f^*(y)\right]. \qquad (47)
\end{aligned}$$

Using (37) and (33), it is easy to see that $\tilde{\phi}_0^* = 0$ and

$$\tilde{\phi}_T^* \leq \frac{1}{2}\left\|x - x^0\right\|_2^2 + \sum_{t=0}^{T-1} \beta_t \left(g(x) + \frac{1}{n}\langle n y^{t+1} - (n-1)y^t, Ax\rangle \right), \quad \forall x \in \mathbb{R}^n.$$

Plugging these to (47) and dropping the term $\|y^T - y\|_2^2$, we obtain:

$$\begin{aligned}
&\frac{1}{2}\left\|x - x^0\right\|_2^2 + \frac{\alpha\beta_0}{2\tau}\|y^0 - y\|_2^2 \qquad (48)\\
\geq &\sum_{t=0}^{T-1} \beta_t \mathbb{E}\left[g(\bar{x}^{t+1}) - g(x) + \frac{1}{n}\langle y, A(\bar{x}^{t+1} - x)\rangle \right]\\
&+ \sum_{t=0}^{T-1} \beta_t \mathbb{E}\left[-\frac{1}{n}\langle n y^{t+1} - (n-1)y^t - y, Ax\rangle + n f^*(y^{t+1}) - (n-1) f^*(y^t) - f^*(y) \right]\\
= &\sum_{t=0}^{T-1} \beta_t \mathbb{E}\left[\tilde{F}(\bar{x}^{t+1}, y) - \tilde{F}(x, y) \right] + \sum_{t=0}^{T-1} \beta_t \mathbb{E}\left[-n\tilde{F}(x, y^{t+1}) + (n-1)\tilde{F}(x, y^t) + \tilde{F}(x, y) \right].
\end{aligned}$$

Now, we choose $(x, y) = (x^*, y^*)$. The first term on the right-hand-side of (48) can be bounded by:

$$\begin{aligned}
\sum_{t=0}^{T-1} \beta_t \mathbb{E}\left[\tilde{F}(\bar{x}^{t+1}, y^*) - \tilde{F}(x^*, y^*) \right] \geq &B_{T-1}\mathbb{E}\left[\tilde{F}(\hat{x}^T, y^*) - \tilde{F}(x^*, y^*) \right]\\
\geq &\frac{B_{T-1}\mu}{2}\mathbb{E}\left[\|\hat{x}^T - x^*\|_2^2 \right], \qquad (49)
\end{aligned}$$

where the μ-strong convexity of $F(\cdot, y^*)$ and the definition of $\hat{x}^T$ are used. By using the fact $\tilde{F}(x^*, y^*) - \tilde{F}(x^*, y) \geq 0$ for any y, we can bound the second term on the right-hand-side of (48) as:

$$\begin{aligned}
&\sum_{t=0}^{T-1} \beta_t \mathbb{E}\left[-n\tilde{F}(x^*, y^{t+1}) + (n-1)\tilde{F}(x^*, y^t) + \tilde{F}(x^*, y^*) \right]\\
= &\sum_{t=0}^{T-1} \beta_t \mathbb{E}\left[n\left(\tilde{F}(x^*, y^*) - \tilde{F}(x^*, y^{t+1}) \right) - (n-1)\left(\tilde{F}(x^*, y^*) - \tilde{F}(x^*, y^t) \right) \right]\\
= &\sum_{t=1}^{T-1} \left(n\beta_{t-1} - (n-1)\beta_t \right) \mathbb{E}\left[\tilde{F}(x^*, y^*) - \tilde{F}(x^*, y^t) \right]\\
&+ n\beta_{T-1}\mathbb{E}\left[\tilde{F}(x^*, y^*) - \tilde{F}(x^*, y^T) \right] - (n-1)\beta_0\left(\tilde{F}(x^*, y^*) - \tilde{F}(x^*, y^0) \right)\\
\geq &-(n-1)\beta_0\left(\tilde{F}(x^*, y^*) - \tilde{F}(x^*, y^0) \right), \qquad (50)
\end{aligned}$$

where the inequality follows from the fact that $n\beta_{t-1} \geq (n-1)\beta_t$. Combining (48), (49) and (50) gives

$$\frac{1}{2}\left\|x^0 - x^*\right\|_2^2 + \frac{\alpha\beta_0}{2\tau}\|y^0 - y^*\|_2^2 + (n-1)\beta_0\left(\tilde{F}(x^*, y^*) - \tilde{F}(x^*, y^0) \right) \geq \frac{B_{T-1}\mu}{2}\mathbb{E}\left[\|\hat{x}^T - x^*\|_2^2 \right],$$

Table 1. Iteration complexities of SDPAD for achieving ϵ-solution accuracy under different settings. Some constants and logarithmic factors are hidden.

	$g(x)$ μ-strongly convex	$g(x)$ non-strongly convex
$f_i(u)$ $(1/\gamma)$-smooth	$\left(n+\bar{R}\sqrt{n/(\mu\gamma)}\right)\log(1/\epsilon)$	$n+\bar{R}\sqrt{n/(\mu\epsilon)}$
$f_i(u)$ non-smooth	$n+\bar{R}\sqrt{n/(\gamma\epsilon)}$	$n+\bar{R}\sqrt{n}/\epsilon$

which leads to the desired result. □

Remark 4. Under Assumption 3.1, the condition number of problem (30) usually defined in stochastic optimization literature (see, e.g., [27]) is $\bar{\kappa}' := \frac{\bar{R}^2}{\mu\gamma}$. Note that $\kappa' \le \bar{\kappa}' \le n\kappa'$. Therefore, Theorem 3.5 implies that the number of iterations needed by SDAPD to achieve ϵ-accuracy is

$$\mathcal{O}\left(\left(n+\sqrt{n\bar{\kappa}'}\right)\log\frac{1}{\epsilon}\right), \tag{51}$$

which matches the lower bound of the complexity of stochastic first-order methods [10]. Moreover, even though $\bar{\kappa}'$ might be larger than κ' in DAPD, (51) still suggests that SDAPD is faster than DAPD, given that each iteration of DAPD is approximately n times more expensive than SDAPD.

From these results, we can conclude that SDAPD is better than regularized dual averaging, the stochastic dual averaging method for minimizing the composite objective function, whose complexity is in the order of $\mathcal{O}(1/\epsilon)$ under the same assumption [23]. Besides, (51) also implies that SDAPD is better than some variance-reduced stochastic methods such as ProxSVRG [24], whose complexity is

$$\mathcal{O}\left(\left(n+\bar{\kappa}'\right)\log\frac{1}{\epsilon}\right),$$

when the condition number $\bar{\kappa}'$ is larger than n. Though some accelerated stochastic methods like Katyusha [1] and SPDC [27] have the same complexity as SDAPD, we will show later that SDAPD is more powerful when the data matrix A is sparse.

Remark 5. Generalization to non-smooth or non-strongly-convex problems. Our results in this section can be extended to non-smooth or non-strongly convex problems easily, by slightly perturbing the primal-dual formulation. When f_i is non-smooth, we can augment f_i^* as $\tilde{f}_i^*(y_i) := f_i^*(y_i) + \frac{\delta_1}{2}(y_i)^2$. While g is non-strongly convex, it can be perturbed as $\tilde{g}(x) := g(x) + \frac{\delta_2}{2}\|x\|_2^2$. Here both δ_1 and δ_2 are small constants that are proportional to the desired solution accuracy ϵ. Following such strategy, we can easily derive the complexities of SDAPD in different seniors, which are presented in Table 1. The derivation is similar to the one in [27], and we omit the details here for succinctness.

4. Efficient Implementation of SDAPD on Sparse Data

In this section, we focus on the case that each vector a_i is a sparse vector so that the data matrix A is also sparse. We show how to efficiently implement SDAPD (Algorithm 2) on problems with sparse A, which can further reduce the per-iteration

complexity of SDAPD from $\mathcal{O}(d)$ to $\mathcal{O}(\rho d)$. Throughout this section, we make the following assumption on function g.

Assumption 4.1. *Assume $g(x)$ is separable, i.e., it can be decomposed as $g(x) = \sum_{j=1}^{d} g_j(x_j)$.*

Here we briefly explain why dual-averaging type algorithm can promote the sparsity. For ease of discussion, we denote $h(x) := f(Ax)$. The dual averaging update is:

$$z^{t+1} = \mathrm{prox}_{B_t g}\left(z^0 - \sum_{k=0}^{t} \beta_k \nabla h(z^k)\right). \tag{52}$$

Note that there is no *direct* dependence between any two consecutive iterates z^t and z^{t+1}. The only place where z^t influences z^{t+1} is in estimating the gradient $\nabla h(z^t)$. In many problems with sparse data, the gradient function $\nabla h(z)$ also possesses sparse structure where only a small portion of coordinates of z^t is required for evaluating $\nabla h(z^t)$. Therefore, dual averaging methods allow lazy sparse update, which only updates the coordinates that will be involved in evaluating the next gradient.

Other types existing stochastic algorithms are incapable of admitting sparse update, except on certain problems with special structures. (See more details in Remark 8). For example, the gradient might not be sparse for some methods like SAGA [4], even when the problem data is sparse. Moreover, some accelerated methods require an extrapolation step which requires to add two dense vectors. In the following, we show how to efficiently implement SDAPD for sparse data.

When implementing SDAPD, we need to keep two auxiliary variables u^t and s^t, which are defined in (35) and (34) respectively. With u^t and s^t on hand, any coordinate of x^t, say x_j^t, can be recovered via

$$x_j^t = \mathrm{prox}_{B_{t-1} g_j}\left(x_j^0 - s_j^t\right)$$

in only $\mathcal{O}(1)$ time, due to the separable assumption of $g(x)$. Similarly, the j-th coordinate of $\bar{x}^{t+1}$ can be computed by

$$\bar{x}_j^{t+1} = \mathrm{prox}_{\eta g_j}\left(x_j^t - \eta u_j^t\right).$$

Note that x^t is only used in the update of $\bar{x}^{t+1}$, while the only role of $\bar{x}^{t+1}$ is for computing the inner product $a_{i_t}^\top \bar{x}^{t+1}$ in (32). This implies that we do not need to evaluate x_j^t and $\bar{x}_j^{t+1}$ when $a_{i_t,j} = 0$. Using this property, the whole iteration of SDAPD can be done in $\mathcal{O}(\|a_{i_t}\|_0)$ computational cost.

Now, the remaining problem is how to update u^{t+1} and s^{t+1} for sparse data. For u^{t+1}, it is straightforward by using (36), which adds a sparse vector a_{i_t} to u^t in each iteration. The real challenge is how to update s^{t+1} in (34), because it is a summation of dense vectors. Here, we present a novel way to sparsify the update of s^{t+1}, by decomposing it into the combination of two sequences. For the ease of discussion, we define

$$\delta^t := \frac{(y_{i_t}^{t+1} - y_{i_t}^t)}{n} \cdot a_{i_t}. \tag{53}$$

Hence, δ^t is a sparse vector if a_{i_t} is sparse. We need to show the following lemma first.

Lemma 4.2. *Consider SDAPD (Algorithm 2) with β_t chosen in the form of $\beta_t = \beta_0\theta^{-t}$ for some $\theta \in (0,1)$. Define two sequences:*

$$v^{t+1} := -\frac{\beta_0\theta}{n(1-\theta)}A^\top y^0 + \sum_{k=0}^{t}\beta_k\left(n - \frac{1}{1-\theta}\right)\delta^k, \text{ and } w^{t+1} := \frac{1}{n(1-\theta)}A^\top y^0 + \frac{1}{1-\theta}\sum_{k=0}^{t}\delta^k.$$

It holds that

$$s^{t+1} := \sum_{k=0}^{t}\frac{\beta_k}{n}A^\top \bar{y}^{k+1} = v^{t+1} + \beta_t w^{t+1} \tag{54}$$

Proof. We prove (54) by induction. we first note two useful relationships:

$$\frac{1}{n}A^\top y^{t+1} = \frac{1}{n}A^\top y^t + \delta^t \tag{55}$$

$$\frac{1}{n}A^\top \bar{y}^{t+1} = \frac{1}{n}A^\top y^t + n\delta^t, \tag{56}$$

which are easy to be obtained from (53), (32) and (33).

When $t = 0$,

$$v^{t+1} = v^1 = -\frac{\beta_0\theta}{n(1-\theta)}A^\top y^0 + \beta_0\left(n - \frac{1}{1-\theta}\right)\delta^0$$

and

$$\beta_t w^{t+1} = \beta_0 w^1 = \frac{\beta_0}{n(1-\theta)}A^\top y^0 + \frac{\beta_0}{1-\theta}\delta^0.$$

By adding these two equations together, we have:

$$v^1 + \beta_0 w^1 = \frac{\beta_0}{n}A^\top y^0 + n\beta_0\delta_0 = \frac{\beta_0}{n}A^\top \bar{y}^1,$$

where the last equality follows from (56). So (54) is proved for $t = 0$.

Now we assume that (54) holds for $t-1$, i.e.,

$$v^t + \beta_{t-1}w^t = \sum_{k=0}^{t-1}\frac{\beta_k}{n}A^\top \bar{y}^{k+1}.$$

Thus

$$v^{t+1} + \beta_t w^{t+1} = \left(v^{t+1} - v^t\right) + \left(\beta_t w^{t+1} - \beta_{t-1}w^t\right) + \sum_{k=0}^{t-1}\frac{\beta_k}{n}A^\top \bar{y}^{k+1}. \tag{57}$$

From (55) and the fact $\beta_{t-1} = \beta_t \theta$, we have:

$$\beta_t w^{t+1} - \beta_{t-1} w^t = \frac{\beta_t}{1-\theta} \delta^t + \frac{\beta_t}{n} A^\top y^t,$$

Hence,

$$\left(v^{t+1} - v^t\right) + \left(\beta_t w^{t+1} - \beta_{t-1} w^t\right) = \beta_t \left(n - \frac{1}{1-\theta}\right) \delta^t + \frac{\beta_t}{1-\theta} \delta^t + \frac{\beta_t}{n} A^\top y^t = \frac{\beta_t}{n} A^\top \bar{y}^{t+1},$$

which is due to (56) again. Combining this equation with (57) proves (54). □

Remark 6. Note that both v^{t+1} and w^{t+1} are actually the summation of sparse vectors δ^t, except the first term $A^\top y^0$. As a result, after computing $A^\top y^0$ at the very beginning of the algorithm, both v^{t+1} and w^{t+1} can be updated in a sparse way. With the help of these two sequences, the whole algorithm is capable of doing sparse update, and thus has only $\mathcal{O}(\rho d)$ per-iteration complexity on average instead of $\mathcal{O}(d)$. In many large scale applications, ρ can be very small like $\rho \approx 10^{-3}$ or even smaller. For example, the well-known DBLP dataset has the sparsity $\rho \approx 2.0 \times 10^{-5}$ [25]. Hence, sparse update can bring great acceleration on such problems.

We now continue the discussion on theoretical complexity of SDAPD. After combining the sparse update technique discussed above, the overall computation cost of SDAPD to achieve ϵ-accuracy becomes

$$\mathcal{O}\left(\rho d \left(n + \sqrt{n\bar{\kappa}'}\right) \log \frac{1}{\epsilon}\right)$$

for strongly convex and smooth problems, if we take both convergence rate and per-iteration computation cost into consideration. This complexity is better than the complexity of existing accelerated stochastic methods like SPDC, namely,

$$\mathcal{O}\left(d \left(n + \sqrt{n\bar{\kappa}'}\right) \log \frac{1}{\epsilon}\right),$$

due to the factor ρ ($0 < \rho \leq 1$).

Remark 7. Lee and Sidford proposed an efficient implementation of accelerated coordinate descent in [11], which shares similar idea of decomposing the updates into two sequences that can be updated efficiently. However, the motivation of their method is different to ours, and our setting is more challenging. Note that the gradient update in [11] is the same as the typical coordinate descent, which naturally requires only $\mathcal{O}(1)$ computation. What they try to avoid is the computation in the extrapolation step of the other coordinates. As a contrast, we do not only have extrapolation step, but the gradients used in update (34) are also the sum of dense vectors. Due to such extra difficulty, our decomposing scheme is different and more complicated than the one in [11].

Remark 8. We point out that a sparse implementation of stochastic SPDC was also proposed in [27], and similar idea for Prox-SVRG can be found in [24]. Such idea can

also be extended to other stochastic methods like ProxSGD and SAGA. However, all these methods implicitly require

$$\text{prox}_g^{(t)} := \underbrace{\text{prox}_g \circ \cdots \circ \text{prox}_g(x)}_{\text{composition of } t \text{ proximal mappings}}$$

can be easily computed in constant time independent of t. This property enables them to ignore the iterations with zero gradients and is the key for their sparse update trick. However, such property is only satisfied by some special $g(x)$, and only examples on simple regularizers $g(x) = \lambda\|x\|_1$ and $g(x) = (\lambda/2)\|x\|_2^2$ are given in their papers. For these two regularizers, it is quite easy to show that

$$\text{prox}_{g_j}^{(t)}(x_j) = \text{prox}_{g_j}^{(t-1)}(x_j) \cdot \frac{1}{1+\lambda} = \cdots = \frac{x_j}{(1+\lambda)^t}$$

for $g(x) = (\lambda/2)\|x\|_2^2$, and

$$\text{prox}_{g_j}^{(t)}(x_j) = \begin{cases} x_j - \text{sign}(x_j) \cdot \lambda t & \text{if } |x_j| \geq \lambda t \\ 0 & \text{otherwise} \end{cases}$$

if $g(x) = \lambda\|x\|_1$. However, as far as we can see, it would be difficult to generalize their method to other regularizers such as KL-divergence, namely,

$$g(x) = \sum_{j=1}^{d} w_j \log \frac{w_j}{x_j},$$

which is commonly used in model-based transfer learning [18]. For this $g(x)$, computing a proximal mapping needs to solve a quadratic equation which does not admit a simple form of solution. Hence, it is hard to compute $\text{prox}_g^{(t)}(x)$ without computing $\text{prox}_g^1(x), \ldots, \text{prox}_g^{(t-1)}(x)$ one by one. As a result, their sparse update method would fail on such regularizer. As a comparison, our method does not rely on such assumption and works with any $g(x)$ as long as it is separable.

5. Numerical Experiments

In this section, we conduct numerical experiments to DAPD and SDAPD and compare their performance with the following relevant existing methods:

- PDHG: primal-dual hybrid gradient method [3]
- APGM: Nesterov's accelerated proximal gradient method [17]
- DA: original dual averaging method [16]
- RDA: regularized dual averaging method [23]
- ProxSGD: proximal stochastic (sub-)gradient method [19]
- ProxSVRG: proximal stochastic variance-reduced gradient method [24]
- SPDC: stochastic primal-dual coordinate method [27]

Note that the first three methods are deterministic methods, while the others are stochastic methods. Besides, PDGH, APGM and SPDC are accelerated methods.

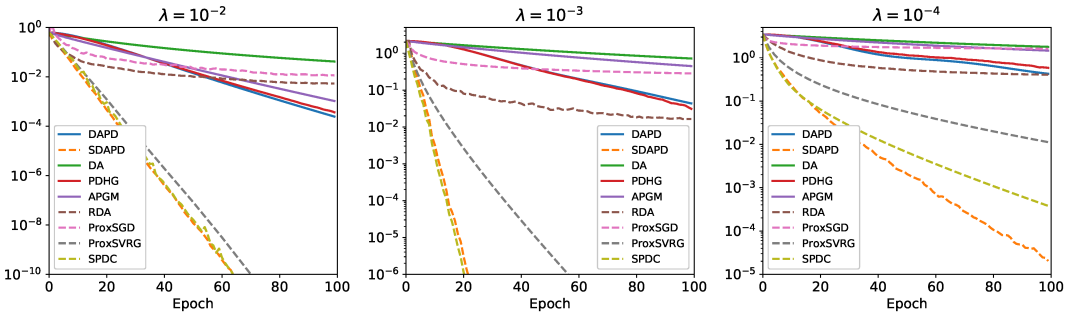

Figure 1. Comparison on synthetic data with different choices of λ. The y-axis is the primal sub-optimality, namely $P(x^t) - P(x^*)$. The solid lines are deterministic methods, and the dashed lines stand for stochastic methods. Here an epoch refers to one iteration for deterministic methods, and n times accesses to the vectors a_i for stochastic methods.

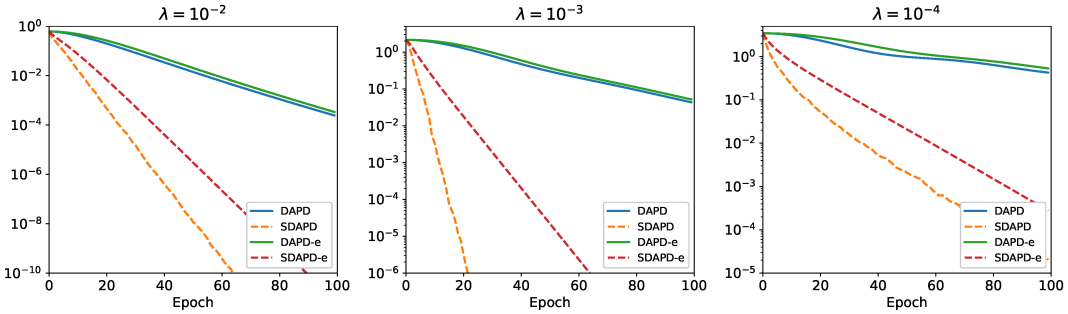

Figure 2. Comparison between ergodic and non-ergodic solutions on synthetic data. The y-axis is the primal sub-optimality, namely $P(x^t) - P(x^*)$. Lines with suffix "-e" stand for ergodic solutions, while others are non-ergodic.

ProxSGD refers to proximal stochastic gradient descent, when we conduct experiments on smooth problems, and refers to stochastic subgradient method if it is applied to non-smooth problems. Though our analysis is based on the ergodic solutions $(\hat{x}^T, \hat{y}^T)$, we mainly report the behavior of the non-ergodic solutions. This is a common practice, because non-ergodic solutions preserve the solution sparsity. For completeness, we also report some comparison of the behavior of the ergodic and non-ergodic solutions in Figure 2.

5.1. Ridge Regression on Synthetic Data

First, we test these algorithms on a ridge regression problem:

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\left(\langle a_i, x\rangle - b_i\right)^2 + \frac{\lambda}{2}\|x\|_2^2$$

with $\lambda > 0$. Note that this problem is smooth and λ-strongly convex. We use synthetic data for this problem. Specifically, we first randomly generate a $x^* \in \mathbb{R}^d$, then each a_i and b_i are independently draw from the following model:

$$b_i = \langle x^*, a_i\rangle + \varepsilon_i \text{ with } a_i \sim \mathcal{N}(0, \Sigma) \text{ and } \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

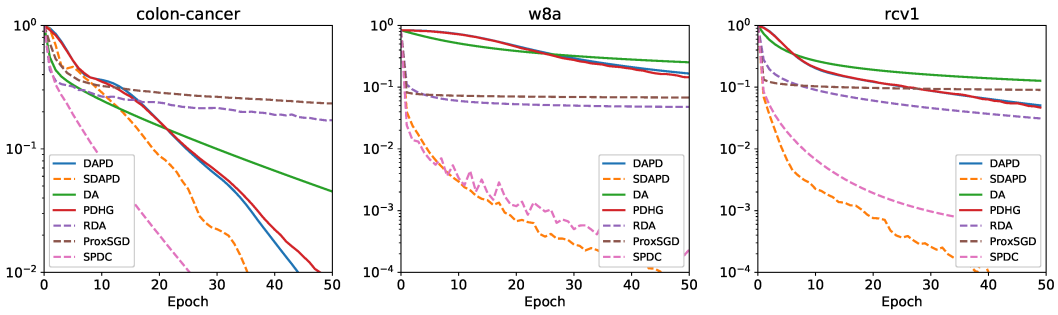

Figure 3. Results on real datasets. The y-axis is the primal sub-optimality.

for some pre-chosen covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$ and constant $\sigma > 0$. In this experiment, we choose $n = d = 1000$. We test the algorithms for different λ, which controls the condition number of the problem. Note that smaller λ leads to larger condition number.

The experiment results are presented in Figure 1. In all three sub-figures, the performances of SDAPD and SPDC are quite close, and are always better than all other methods. Besides, when $\lambda = 10^{-2}$, ProxSVRG also performs well, but it soon becomes inferior than SDAPD and SPDC when λ gets smaller, since ProxSVRG is not an accelerated method. We also found that DAPD method performs similarly as PDGH, and is always much better than the other two deterministic methods: APGM and DA, though it is slower than some stochastic methods. Besides, when the condition number becomes larger, the performance difference between deterministic methods and stochastic methods becomes more prominent.

Although the ergodic solutions of dual-averaging-type methods are rarely used in practice, we still report its behavior in Figure 2 for completeness. Figure 2 shows that the ergodic solutions converge slower than the non-ergodic solutions.

5.2. Classification via SVM on Real Datasets

In this part, we test the algorithms on the binary classification task via support vector machine (SVM):

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max\left\{1 - \langle b_i a_i, x\rangle\,,\, 0\right\} + g(x).$$

Table 2. Summary of datasets

Dataset	n	d	ρ
`colon-cancer`	62	2,000	100%
`w8a`	49,749	300	3.88%
`rcv1`	20,242	47,236	0.16%

Table 3. Per-epoch running time of each method in seconds

Methods	`colon-cancer` ($\times 10^{-3}$)	`w8a` ($\times 10^{-2}$)	`rcv1` ($\times 10^{-2}$)
DAPD	1.0	5.3	4.2
SDAPD	5.2	17.5	19.8
PDHG	1.0	5.7	4.0
DA	1.0	5.5	3.6
RDA	2.7	7.4	6.0
ProxSGD	2.9	30.3	1932
SPDC	2.7	46.3	2125

Here we choose $g(x)$ to be the Huber's regularization, which is defined as: $g(x) = \sum_{j=1}^{d} g_j(x_j)$ with

$$g_j(x_j) = \begin{cases} \lambda\big(|x_j| - \frac{\lambda}{4\mu}\big) & \text{if } |x_j| \geq \frac{\lambda}{2\mu}, \\ \mu x_j^2 & \text{otherwise.} \end{cases} \tag{58}$$

Huber's regularization can also help the model to avoid over-fitting just like the squared-ℓ_2-norm, but it is statistically more robust than the latter one [26]. As far as we know, it would be hard for ProxSGD and SPDC to have sparse update with such $g(x)$. For this experiment, we fix the parameters as $\lambda = 10^{-4}$ and $\mu = 1$ in Huber's regularization. Besides, it should be noted that our objective function is non-smooth in this case. Since APGM and ProxSVRG are unable to deal with such kind of objective, they are not tested for this problem. We use real datasets in this experiment. The dataset information is summarized in Table 2. `w8a` and `rcv1` are sparse datasets.

The experiment results are presented in Figure 3. The results are similar to the ones for ridge regression. We observe that SDAPD performs better than all other methods, except that it falls behind SPDC on `colon-cancer`. Again, the performances of DAPD and PDHG are very close, but they are much better than the other deterministic method DA. Only thing interesting to note here is that the performance of DAPD is close to SDAPD on `colon-cancer`. It is because this dataset has a relatively small n, thus deterministic methods and stochastic methods do not make too much difference in their convergence rates.

We also report the per-epoch running time of each algorithm in Table 3. We see that deterministic methods DAPD, PDGH and DA are always the fastest, since they can do updates in batch with highly-optimized matrix-vector operations. We can also find that ProxSGD and SPDC are quite time-consuming on `w8a` and `rcv1` datasets because they are unable to do sparse updates, while our SDAPD overcomes this issue with the help of the sparse update strategy introduced in Section 4 and therefore has much less computational cost on sparse data. However, such strategy requires to maintain some auxiliary variables, resulting more computational time to SDAPD than RDA, and even costs more time than ProxSGD and SPDC on dense data. Of course, SDAPD can be further improved by discarding the sparse update strategy on dense data. But we do not adopt this here. Instead, we implement SDAPD in a uniform way to keep the experiments simple.

Overall, by taking both convergence rate and per-epoch computation time into account, SDAPD is the best one among all tested algorithms.

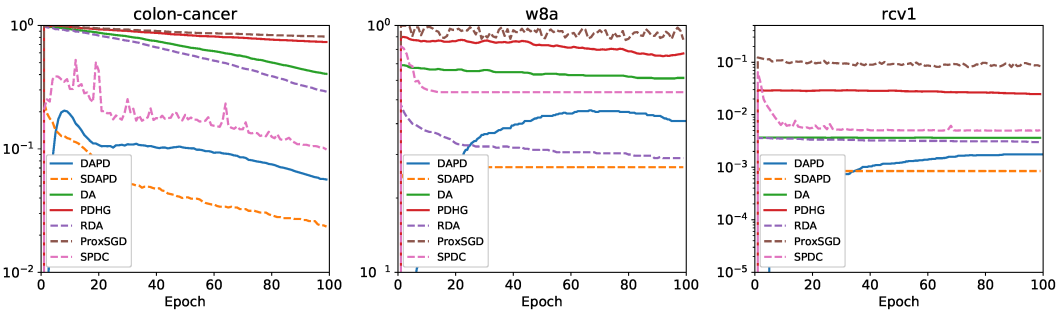

Figure 4. Proportion of non-zeros in the generated non-ergodic solutions.

5.3. Comparison on Solution Sparsity

In this part, we focus on the same setting as the previous part. However, we change the regularizer to $g(x) = \lambda\|x\|_1$ to induce sparse solution, so that we can observe the influence of different optimization methods to solution sparsity. Again, we fix $\lambda = 10^{-4}$ on all the datasets.

The results are presented in Figure 4, which show that our DAPD and SDAPD can produce sparser solutions than most baselines. The only exception is RDA, which is comparable with DAPD and SDAPD on the `w8a` dataset. This is expected because RDA is known to promote solution sparsity. Moreover, we found that stochastic algorithm SDAPD always outperforms RDA on the tested problems. We conjecture it is because of the increasing primal step sizes $\{\beta_t\}$ in our algorithms that make regularization effects even stronger.

6. Conclusion

In this paper, we proposed a dual-averaging primal-dual method (DAPD), which combines the idea of dual averaging and primal-dual method, and can solve a wide range of optimization problems with composite convex objective. Our analysis shows that DAPD has optimal convergence rates in several different settings. We also proposed a stochastic version of DAPD (SDAPD) for solving convex problems with a finite-sum objective. A novel way is proposed to efficiently implement SDAPD for sparse data. We demonstrated the superiority of our methods by comparing them with several existing methods on standard machine learning tasks.

Acknowledgements

The authors are grateful to two anonymous referees for providing insightful and constructive comments that greatly improved the presentation of this paper. The research of S. Ma was supported in part by a startup package in the Department of Mathematics at University of California, Davis.

References

[1] Z. Allen-Zhu, *Katyusha: The first direct acceleration of stochastic gradient methods*, in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2017, pp. 1200–1205.

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

[3] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of mathematical imaging and vision 40 (2011), pp. 120–145.

[4] A. Defazio, F. Bach, and S. Lacoste-Julien, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, in *Advances in neural information processing systems*. 2014, pp. 1646–1654.

[5] L. Gao, J. Song, X. Liu, J. Shao, J. Liu, and J. Shao, *Learning in high-dimensional multimedia data: the state of the art*, Multimedia Systems 23 (2017), pp. 303–313.

[6] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in *Advances in neural information processing systems*. 2013, pp. 315–323.

[7] S. Kakade, S. Shalev-Shwartz, and A. Tewari, *On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization*, Unpublished Manuscript (2009).

[8] G. Korpelevich, *Extrapolation gradient methods and relation to modified lagrangeans. ekonomika i matematicheskie metody, 19: 694–703, 1983*, Russian; English translation in Matekon .

[9] G. Korpelevich, *The extragradient method for finding saddle points and other problems*, Matecon 12 (1976), pp. 747–756.

[10] G. Lan and Y. Zhou, *An optimal randomized incremental gradient method*, Mathematical programming (2017), pp. 1–49.

[11] Y.T. Lee and A. Sidford, *Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems*, in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 2013, pp. 147–156.

[12] H.B. McMahan, *Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization*, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011, pp. 525–533.

[13] H.B. McMahan, *A survey of algorithms and analysis for adaptive online learning*, The Journal of Machine Learning Research 18 (2017), pp. 3117–3166.

[14] H.B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, *et al.*, *Ad click prediction: a view from the trenches*, in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1222–1230.

[15] T. Murata and T. Suzuki, *Doubly accelerated stochastic variance reduced dual averaging method for regularized empirical risk minimization*, in *Advances in Neural Information Processing Systems*. 2017, pp. 608–617.

[16] Y. Nesterov, *Primal-dual subgradient methods for convex problems*, Mathematical programming 120 (2009), pp. 221–259.

[17] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, Vol. 87, Springer Science & Business Media, 2013.

[18] S.J. Pan and Q. Yang, *A survey on transfer learning*, IEEE Transactions on knowledge and data engineering 22 (2009), pp. 1345–1359.

[19] O. Shamir and T. Zhang, *Stochastic gradient descent for non-smooth optimization:*

Convergence results and optimal averaging schemes, in *International Conference on Machine Learning*. 2013, pp. 71–79.

[20] C. Tan, T. Zhang, S. Ma, and J. Liu, *Stochastic Primal-Dual Method for Empirical Risk Minimization with $\mathcal{O}(1)$ Per-Iteration Complexity*, in *Advances in Neural Information Processing Systems*. 2018, pp. 8376–8385.

[21] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Royal. Statist. Soc B. 58 (1996), pp. 267–288.

[22] P. Tseng, *On accelerated proximal gradient methods for convex-concave optimization*, submitted to SIAM Journal on Optimization (2008).

[23] L. Xiao, *Dual averaging methods for regularized stochastic learning and online optimization*, Journal of Machine Learning Research 11 (2010), pp. 2543–2596.

[24] L. Xiao and T. Zhang, *A proximal stochastic gradient method with progressive variance reduction*, SIAM Journal on Optimization 24 (2014), pp. 2057–2075.

[25] J. Yang and J. Leskovec, *Defining and evaluating network communities based on ground-truth*, Knowledge and Information Systems 42 (2015), pp. 181–213.

[26] O. Zadorozhnyi, G. Benecke, S. Mandt, T. Scheffer, and M. Kloft, *Huber-norm regularization for linear prediction models*, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 714–730.

[27] Y. Zhang and L. Xiao, *Stochastic primal-dual coordinate method for regularized empirical risk minimization*, The Journal of Machine Learning Research 18 (2017), pp. 2939–2980.