electronics

MDPI

Article

ASPDC: Accelerated SPDC Regularized Empirical Risk Minimization for Ill-Conditioned Problems in Large-Scale Machine Learning

Haobang Liang [1], Hao Cai [2], Hejun Wu [3,*], Fanhua Shang [4], James Cheng [5] and Xiying Li [6]

[1] School of Biomedical Engineering, Sun Yat-sen University, Guangzhou 510006, China; lianghb6@mail2.sysu.edu.cn
[2] College of Engineering, Shantou University, Shantou 515041, China; haocai@stu.edu.cn
[3] Department of Computer Science, Sun Yat-sen University, Guangzhou 510006, China
[4] School of Artificial Intelligence, Xidian University, Xi'an 710071, China; fhshang@xidian.edu.cn
[5] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China; jcheng@cse.cuhk.edu.hk
[6] School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510006, China; stslxy@mail.sysu.edu.cn
* Correspondence: wuhejun@mail.sysu.edu.cn

Abstract: This paper aims to improve the response speed of SPDC (stochastic primal–dual coordinate ascent) in large-scale machine learning, as the complexity of per-iteration of SPDC is not satisfactory. We propose an accelerated stochastic primal–dual coordinate ascent called ASPDC and its further accelerated variant, ASPDC-i. Our proposed ASPDC methods achieve a good balance between low per-iteration computation complexity and fast convergence speed, even when the condition number becomes very large. The large condition number causes ill-conditioned problems, which usually requires many more iterations before convergence and longer per-iteration times in data training for machine learning. We performed experiments on various machine learning problems. The experimental results demonstrate that ASPDC and ASPDC-i converge faster than their counterparts, and enjoy low per-iteration complexity as well.

Keywords: stochastic optimization; machine learning; empirical risk minimization; coordinate ascent algorithm; primal–dual algorithm; strongly convex and smooth

Citation: Liang, H.; Cai, H.; Wu, H.; Shang, F.; Cheng J.; Li X. ASPDC: Accelerated SPDC Regularized Empirical Risk Minimization for Ill-Conditioned Problems in Large-Scale Machine Learning. *Electronics* **2022**, *11*, 2382. https://doi.org/10.3390/electronics11152382

Academic Editor: Ahmad Taher Azar

Received: 30 June 2022
Accepted: 26 July 2022
Published: 29 July 2022

1. Introduction

In this paper, we consider a composite convex optimization problem, Regularized Empirical Risk Minimization (RERM), that can be solved by SPDC [1]. Our goal is to use our proposed ASPDC find the approximate solution of the following optimization problem:

$$\min_{w\in\mathbb{R}^d}\{P(w) = \frac{1}{n}\sum_{i=1}^{n}\phi_i(y_i, w^T x_i, b) + g(w)\} \tag{1}$$

where $x_i \in \mathbb{R}^d$ is a feature vector, y_i is the corresponding label in a machine learning task, $\{(x_i, y_i)\}$, $i = 1, 2, \ldots, n$ are n samples in the dataset, ϕ_i is the proper convex function of the linear predictor $w^T x_i$, and $g(w)$ the simple convex regularization function.

RERM is one of the central problems in machine learning. It is now prevalent in the data mining and machine learning domain. More background information on RERM can be found in [2]. The following are four examples of RERM:

1. Linear SVM, where $\phi_i(y_i, w^T x_i, b) = \max\{0, 1 - y_i(w^T x_i + b)\}$, $g(w) = \frac{\lambda}{2}||w||_2^2$
2. Ridge Regression, where $\phi_i(y_i, w^T x_i, b) = \frac{1}{2}(y_i - (w^T x_i + b))^2$, $g(w) = \frac{\lambda}{2}||w||_2^2$
3. Lasso, where $\phi_i(y_i, w^T x_i, b) = \frac{1}{2}(y_i - (w^T x_i + b))^2$, $g(w) = \lambda||w||_1$

4. Logistic Regression, where $\phi_i(y_i, w^T x_i, b) = log(1 + \exp(-y_i(w^T x_i + b)))$, $g(w) = \frac{\lambda}{2}||w||_2^2$

Here, we focus on the scenario in which the number of samples n is very large, as the per-iteration complexity of SPDC is intolerable in this scenario. Computing a full gradient becomes extremely expensive in terms of time and space costs. Therefore, RERM algorithms with a lower per-iteration complexity are more attractive in large-scale machine learning applications.

General optimization methods to the RERM problem using gradients are categorized into two types, namely, first-order and second-order. Second-order methods such as the Newton algorithm employ a Hessian matrix at each iteration to decrease the objective value. The disadvantage of these second-order methods is that both obtaining and using a Hessian matrix is computationally expensive. On the other hand, while first-order optimization schemes are lightweight in gradient computation, they may converge slowly [3,4].

Among the algorithms for solving the RERM problem, we are more interested in dual algorithms such as stochastic dual coordinate ascent-SDCA, as the dual-gap is a clearer stopping criterion than gradients. In addition, they are capable of handling non-differentiable primal optimal functions more easily [5]. SDCA is a first-order optimization method and is widely used in the current machine learning domain. Dual coordinate methods have been implemented in open machine learning libraries [4].

The dual methods do not solve the primal problem directly. Instead, they solve the dual or saddle point problem of the primal problem. The corresponding dual problem of the primal problem in Equation (1) is formulated as follows:

$$\max_{\alpha \in \mathbb{R}^n} \{D(\alpha) = \frac{1}{n} \sum_{i=1}^{n} -\phi_i^*(\alpha_i) - g^*(-\frac{1}{n} \sum_{i=1}^{n} \alpha_i x_i)\} \tag{2}$$

where $g^*(u) = \max_{w \in \mathbb{R}^d} \{w^T u - g(w)\}$ and ϕ_i^* are the convex conjugate functions of g and ϕ_i, respectively. Due to the structure of this dual problem, coordinate ascent methods can be more efficient than full gradient methods [4,6,7] .

In the stochastic dual coordinate ascent method (SDCA) [5], a dual coordinate α_i is picked randomly at each iteration and then updated to increase the dual objective value. This helps SDCA to reach a low per-iteration computational complexity. Nevertheless, the convergency speed of SDCA becomes much slower as the condition number grows. A large condition number leads to an ill-conditioned problem. An ill-conditioned scenario refers to a case in which a small change in one of the values of the coefficient matrix causes a large change in the solution vector [8–11]. Hence, SDCA is not applicable to large-scale data processing in ill-conditioned scenarios. Unfortunately, many traning tasks involving large-scale data involve ill-conditioned scenarios. Ill-conditioned problems are particularly common in mathematics and geosciences [12].

Paper Organization. The rest of this paper is organized as follows. In Section 2, we describe related works.

In Section 3, we describe the relevant assumptions and preliminaries.

In Section 4, we discuss the accelerated stochastic primal–dual coordinate method. In this section, we present ASPDC in Algorithm 1 and its convergence analysis for the saddle point problem in Equation (3).

In Section 5, we extend ASPDC to ill-conditioned problems, in particular, those in which $\lambda \leq \frac{4}{n\gamma}$. Our proposed extension method is called ASPDC-i, where i means "for ill-conditioned problems".

In Section 6, we evaluate the performance of our proposed ASPDC algorithms with several state-of-art algorithms for solving machine learning problems, then discuss the experimental results.

In Section 7, we conclude the paper and discuss potential avenues for future work.

2. Related Work

Shalev-Shwartz and Zhang [13] developed an accelerated proximal stochastic dual coordinate ascent method (ASDCA), which converges faster than traditional methods when the condition number is large (Table 1). ASDCA can be regarded as a variant of a proximal point algorithm equipped with Nesterov's accelerated technique [14–16]. ASDCA uses an inner–outer iteration procedure, where the outer loop is a minimization of an auxiliary problem with a regularized quadratic term. Then, the proximal SDCA starts to solve the auxiliary problem with a customized precision. At the end of each outer loop, Nesterov's accelerated update is performed on the primal variable w. Nonetheless, ASDCA requires λ to be limited to a range of low-level values, for example, $\lambda \leq \frac{R^2}{10n\gamma}$, where γ is the smooth parameter of ϕ_i, n is the number of samples, and $R^2 = \max_i ||x_i||_2^2$.

Studies have extended the inner–outer iteration method in order to derive more general accelerated proximal-point algorithms, e.g., Catalyst, [17,18]. Theoretically, one can replace the inner-loop proximal SDCA algorithm using other algorithms, such as SVRG [19] and Prox-SVRG [20], to obtain the same overall complexity concerning the number of outer loops.

More recently, Zhang and Xiao [1,21] proposed a stochastic primal–dual coordinate (SPDC) method to solve the RERM problem defined in Equation (1). SPDC achieves a faster convergence rate in reducing the dual-gap than ASDCA and other dual methods in general optimization problems with condition numbers that are not very large. The per-iteration computation complexity of SPDC is much higher than ASDCA and SDCA. Theoretically, the per-iteration complexity of SPDC is $\mathcal{O}(d)$. However, due to the auxiliary variable update and the momentum item, SPDC requires much more time to process one pass of a dataset, as verified in our experiments. When the condition number is large, the SPDC per-iteration computation complexity of SPDC is intolerable, which makes SPDC inapplicable to large-scale data processing. Our experiments verified that SPDC is more time-consuming than ASDCA and other low per-iteration complexity methods. Moreover, the dual-gap of SPDC is much larger when the data are sparse and have high dimensionality.

The above issue leads to the following key question: "Can we design an algorithm with both a low per-iteration complexity and a fast convergence rate, especially for ill-conditioned scenarios in large-scale data processing?" We propose the ASPDC and ASPDC-i algorithms as the answer to this question. ASPDC methods have the following three advantages:

- Simple structure at each iteration. In comparison with SPDC or other accelerated variants, ASPDC does not need to keep track of any other auxiliary variables; it only maintains the primal and dual variable. Each iteration only involves a dual update and primal update. This design makes its per-iteration complexity much lower than SPDC and other variants. The simple iteration design makes it easy to be implemented as well.
- Short running time. Our experiments show that to reach the same precision, our methods need far less time and fewer epochs (numbers of passes through the entire data) to satisfy the stop condition.
- Theoretical guarantee. ASPDC adopts Nesterov's estimation technique [22,23]. We present a new proof onf the convergence of proposed methods.

Table 1. Abbreviations used in this study.

Complete Name	Abbreviation
Stochastic primal-dual coordinate ascent	SPDC
Stochastic dual coordinate ascent method	SDCA
Accelerated stochastic primal-dual coordinate ascent	ASPDC
Extended ASPDC to the ill-conditioned problem	ASPDC-i
Accelerated stochastic dual ascent	ASDCA

3. Assumptions and Preliminary

Throughout this paper, the standard Euclidean is denoted as an equation such as $||w||_2 = \sqrt{\sum_i |w_i|^2}$. We use E to denote the expectation that is taken with respect to the randomness of α_i. For the sake of convenience, we use the new notation $x_i \leftarrow (x_i^T, 1)^T$, $w \leftarrow (w^T, b)^T$. Without loss of generality, we continue to assume $w \in \mathbb{R}^d$, $x_i \in \mathbb{R}^d$. Then, we make the following assumptions to clearly specify the problem in Equation (1) as follows:

Assumption 1. *Each ϕ_i is lower semi-continuous and convex, and its derivative is $\frac{1}{\gamma}$-Lipschitz continuous (or equivalently: ϕ_i is $\frac{1}{\gamma}$-smooth), i.e., there exist $\gamma > 0$ such that $|\phi_i'(a) - \phi_i'(b)| \leqslant \frac{1}{\gamma}|a - b| \quad \forall a, b \in \mathbb{R} \quad i = 1, 2, \ldots, n$.*

It is widely known that Assumption 1 implies that ϕ_i^* is γ-strongly convex (see Theorem 4.2.2 in the convex fundamental book [24]).

Assumption 2. *The primal function $P(w)$ is λ-strongly convex: There exists $\lambda > 0$ such that $\forall w_1, w_2 \in \mathbb{R}^d$,*

$$P(w_1) \geq P(w_2) + \nabla P(w_2)^T (w_1 - w_2) + \frac{\lambda}{2}||w_1 - w_2||_2^2$$

The convexity of $P(w)$ may come from either ϕ_i or $g(w)$ or both. For instance, if $g(w) = \frac{\lambda}{2}||w||_2^2$, Assumption 2 holds.

Assumption 3. $||x_i||_2 \leqslant 1, \forall i = 1, 2, \ldots, n$.

Assumption 3 is not a strict one, as when data are normalized, Assumption 3 holds.

Under the three assumptions above, the RERM problem defined in Equation (1) can be rewritten as the following convex–concave saddle point problem [1]:

$$\min_{w \in \mathbb{R}^d} \max_{\alpha \in \mathbb{R}^n} \{f(w, \alpha) = \frac{1}{n} \sum [\alpha_i w^T x_i - \phi_i^*(\alpha_i)] + g(w)\} \tag{3}$$

where $\phi_i^*(\alpha_i) = \sup_{s \in \mathbb{R}} \{s\alpha_i - \phi_i(s)\}$ is a convex conjugation function of ϕ_i. Lemma 1 demonstrates the relationship between the primal problem of Equation (1) with the problem of Equation (3).

Lemma 1. *Let $w^* = \arg\min_{w \in \mathbb{R}^d} P(w)$ and $\alpha^* = \arg\max_{\alpha \in \mathbb{R}^n} D(\alpha)$, then we have*

(1) $P(w) = \max_{\alpha \in \mathbb{R}^n} f(w, \alpha)$

(2) $D(\alpha) = \min_{w \in \mathbb{R}^d} f(w, \alpha)$

(3) There exists a unique solution (w^*, α^*) such that $P(w^*) = D(\alpha^*) = f(w^*, \alpha^*)$.

Proof. Presented in Appendix A. □

Lemma 1 implies that we can calculate the optimal solution of the primal problem in Equation (1) by solving the saddle point problem in Equation (3).

4. Accelerated Stochastic Primal–Dual Coordinate Method

In this section, we present ASPDC in Algorithm 1 and its convergence analysis for the saddle point problem in Equation (3).

Each iteration in ASPDC can be divided into two steps: the dual update step and the primal update step. The dual update step is executed first. As shown in lines 4–6 of Algorithm 1, a dual coordinate, α_i, is picked randomly and updated to increase the objective value of $f(w, \alpha)$ while keeping the primal variable w and other $\alpha_j (j \neq i)$ fixed. Then, the

primal update step is executed later. As shown in line 7 of Algorithm 1, the primal variable w is updated to decrease the objective value of $f(w, \alpha)$ while keeping $\alpha_j (j = 1, 2, \ldots, n)$ fixed.

The update of the dual variable α is extremely simple. It can be simplified as a univariate optimal problem, which makes its per-iteration complexity much lower than traditional SPDC algorithms. Specifically, the local update of dual variable α_i is

$$\begin{aligned} \Delta\alpha_i^* &= \arg\max_{\Delta\alpha_i \in \mathbb{R}} f(w, \alpha + \Delta\alpha_i e_i) \\ &= \arg\max_{\Delta\alpha_i \in \mathbb{R}} (\Delta\alpha_i x_i^T w^{(t)} - \phi_i^*(\alpha_i^{(t)} + \Delta\alpha_i)), \end{aligned} \tag{4}$$

where $e_i \in \mathbb{R}^n$ is a unit vector with the $i - th$ element being one.

The update of primal variable w is shown in Equation (5) as follows:

$$w^* = \arg\min_{w \in \mathbb{R}^d} f(w, \alpha^{(t+1)}) \tag{5}$$

$$= \arg\min_{w \in \mathbb{R}^d} \{(\frac{1}{n} \sum_{i=1}^{n} \alpha_i^{(t+1)} x_i)^T w + g(w)\} \tag{6}$$

$$= \arg\max_{w \in \mathbb{R}^d} \{(-\frac{1}{n} \sum_{i=1}^{n} \alpha_i^{(t+1)} x_i)^T w - g(w)\} \tag{7}$$

$$= \nabla g^*(-\frac{1}{n} \sum_{i=1}^{n} \alpha_i^{(t+1)} x_i), \tag{8}$$

where the last equation is derived from the conjugation sub-gradient theorem in [25]. In this way, we turn the optimization process into a derivative operation of $g^*(w)$. For instance, if $g(x) = \frac{\lambda}{2}||w||_2^2$ the update of primal variable can be written as $w^{(t+1)} = -\frac{1}{\lambda n} \sum_{i=1}^{n} \alpha_i^{(t+1)} x_i$.

We compare the complexity of SPD1, SPD1-VR, and SVRG [19] with our methods in Table 2. In Table 2, r is the maximum number of non-zero elements in each sample, S is the number of non-zero elements in the whole data sets, d is the dimension of the dataset, and n the number of data samples. Usually, S is much smaller than nd when the data are sparse and high-dimensional. Apparently, in most large-scale data applications the data sets are sparse have high dimensionality, i.e., most of the attributes are zeros. At each iteration, SPD1 and SPD1-VR choose x_{ij} (the j-th value of sample x_i) to update the primal variable and dual variable regardless of whether x_{ij} is 0 or not. This method enables the per-iteration complexity of SPD1 and SPD1-VR to be reduced to $\mathcal{O}(1)$. However, their complexity of pass-through data is $\mathcal{O}(nd)$, which is the same as SVRG. In contrast, ASPDC will not execute the update if $x_{ij} = 0$. Thus, the complexity of its pass-through data is $\mathcal{O}(S)$, which is much lower than SPD1 and SVRG when the data are sparse and high-dimensional.

Table 2. Complexity comparison of per-iteration and pass through data.

	Per-Iteration	**Pass through Data**
ASPDC, ASPDC-i	$\mathcal{O}(r)$	$\mathcal{O}(S)$
SPD1, SPD1-VR	$\mathcal{O}(1)$	$\mathcal{O}(nd)$
SVRG [19]	$\mathcal{O}(d)$	$\mathcal{O}(nd)$

There are two major differences between SDCA and ASPDC, as follows. First, SDCA tries to solve the dual problem, while ASPDC tries to solve a saddle point problem. Second, the dual update of ASPDC is significantly simpler than the update of SDCA. The dual

update of SDCA is shown in (9). In comparison with that of ASPDC in Equation (4), the dual update of SDCA involves the additional computation of $\frac{1}{2\lambda n}||x_i||_2^2(\Delta\alpha_i)^2$:

$$\Delta\alpha_i^* = \arg\max_{\Delta\alpha_i\in\mathbb{R}}(-\Delta\alpha_i x_i^T w^{(t)} - \phi_i^*(-\alpha_i^{(t)} - \Delta\alpha_i)) + \frac{1}{2\lambda n}||x_i||_2^2(\Delta\alpha_i)^2 \quad (9)$$

We use the dual-gap metric as the stopping criterion, as shown in line 9 of Algorithm 1. The dual-gap is calculated by $P(w) - D(\alpha)$, and it is sufficient to say that $|P(w) - P(w^*)| \leq \epsilon$ if $P(w) - D(\alpha) \leq \epsilon$, as $|P(w) - p(w^*)| \leq P(w) - D(\alpha) \leq \epsilon$. This stopping criterion is easier to implement than the other criteria, e.g., $|P(w) - P(w^*)| \leq \epsilon$. This is for the reason that w^* is not known in advance in real-world machine learning applications.

Algorithm 1 ASPDC

1: Input $f(w, \alpha)$, $\alpha^{(0)}$, ϵ
2: Initialize $w^{(0)} = \nabla g^*(-\frac{1}{n}\sum_{i=1}^n \alpha_i^{(0)} x_i)$
3: **for** $t = 0, 1, 2, \ldots$ **do**
4: pick $i \in \{1, 2, \ldots, n\}$ under uniform distribution.
5: $\Delta\alpha_i^* = \arg\max_{\Delta\alpha_i\in\mathbb{R}}(\Delta\alpha_i x_i^T w^{(t)} - \phi_i^*(\alpha_i^{(t)} + \Delta\alpha_i))$
6: $\alpha^{(t+1)} = \alpha^{(t)} + \Delta\alpha_i^* e_i$
7: $w^{(t+1)} = \nabla g^*(-\frac{1}{n}\sum_{i=1}^n \alpha_i^{(t+1)} x_i)$
8: **end for**
9: Stop condition: $P(w^{(T)}) - D(\alpha^{(T)}) \leq \epsilon$
Output $w^{(T)}$, $\alpha^{(T)}$, $P(w^{(T)}) - D(\alpha^{(T)})$

In the rest of this section, we show the proof for ASPDC's convergence. We first present the following lemma.

Lemma 2. *On the basis of Assumptions 1–3, let $w^{(t)}$ and $\alpha^{(t)}$ be the sequence produced by ASPDC and let $g(w) = \frac{\lambda}{2}||w||_2^2$. $\forall\lambda \geq \frac{4}{n\gamma}$; then, we have:*

$$E(P(w^{(t)}) - D(\alpha^{(t)})) \leq 2n(1 - \frac{1}{2n})^t(P(w^{(0)}) - D(\alpha^{(0)})) \quad (10)$$

Proof. The detailed proof can be found in the Appendix. In the proof, we assume that $g(w) = \frac{\lambda}{2}||w||_2^2$ for convenience. Therefore, the theory only works for l2 regularization. The extension to l1 regularization is a topic for future work.

The skeleton of the proof in the Appendix can be described using the following three steps:

First, we obtain

$$E(D(\alpha^*) - D(\alpha^{(t)})) \leq (1 - \frac{1}{2n})^t(D(\alpha^*) - D(\alpha^{(0)})).$$

Second, we have

$$\begin{aligned}
&\frac{1}{2n}E(P(w^{(t)}) - D(\alpha^{(t)})) \\
&\leq E(D(\alpha^{(t+1)}) - D(\alpha^{(t)})) \\
&\leq E(D(\alpha^{(t+1)}) - D(\alpha^*) + D(\alpha^*) - D(\alpha^{(t)})) \\
&\leq D(\alpha^*) - D(\alpha^{(t)}) - E(D(\alpha^*) - D(\alpha^{(t+1)}) \\
&\leq D(\alpha^*) - D(\alpha^{(t)})
\end{aligned}$$

Finally, using the weak duality we can obtain

$$E(P(w^{(t)}) - D(\alpha^{(t)})) \leq 2n(1 - \frac{1}{2n})^t(P(w^{(0)}) - D(\alpha^{(0)})).$$

□

Theorem 1. *The total number of iterations needed to achieve the expected duality gap of* $E(P(w^{(t)}) - D(\alpha^{(t)})) \leq \epsilon$ *is*

$$t \geq 2n \log(2n(P(w^{(0)}) - D(\alpha^{(0)}))\frac{1}{\epsilon})$$

Proof. Using Lemma 2, we can obtain

$$E(P(w^{(t)}) - D(\alpha^{(t)})) \leq 2n \exp(\frac{-t}{2n})(P(w^{(0)}) - D(\alpha^{(0)})), \tag{11}$$

where, in the inequality, we use the fact that $(1 - \frac{1}{2n})^t \leq \exp(\frac{-t}{2n})$. Let $2n \exp(\frac{-t}{2n})(P(w^{(0)}) - D(\alpha^{(0)})) \leq \epsilon$; then, we finally obtain $t \geq 2n \log(2n(P(w^{(0)}) - D(\alpha^{(0)}))\frac{1}{\epsilon})$. □

As shown by Equation (11), the complexity of ASPDC is $\mathcal{O}(n \log(n\frac{1}{\epsilon}))$, In contrast, the complexity of SVRG is $\mathcal{O}(d(n + \kappa) \log(1/\epsilon))$ and the complexity of SPDC is $\mathcal{O}(d(n + \sqrt{n\kappa}) \log(1/\epsilon))$.

5. ASPDC for Ill-Conditioned Problems

According to convex theory [16], the value $Q_f = L/\mu$ is called the condition number of function f if f is $L - smooth$ and $\mu - strongly\ convex$. Under Assumptions 1–3, the condition number of the primal function in Equation (1) is $(1 + \gamma\lambda)/\lambda = \frac{1}{\lambda\gamma} + 1$. Suppose λ becomes lower; then, the condition number, Q_f, will be larger. When $Q_f \gg 1$, the problem f is called ill-conditioned.

In this section, we extend ASPDC to the ill-conditioned problem, especially when $\lambda \leq \frac{4}{n\gamma}$. The extension method is called ASPDC-i, in which the suffix i means "for ill-conditioned problems".

As shown in Algorithm 2, the procedure of ASPDC-i can be divided into epochs, indexed $s = 1, 2, 3, \ldots, S$. Each epoch uses ASPDC to solve the following problem with a decreasing precision parameter ξ_s:

$$\min_{w \in \mathbb{R}^d} \max_{\alpha \in \mathbb{R}^n} \widetilde{f}_s(w, \alpha) = \frac{1}{n}\sum_{i=1}^{n}[\alpha_i w^T x_i - \phi_i^*(\alpha_i)] + \widetilde{g}(w) \tag{12}$$

where $\widetilde{g}(w) = g(w) + \frac{\kappa}{2}||w||_2^2 - \kappa w^T \widetilde{w}^s$, $\kappa \in \mathbb{R}$ is a constant throughout the procedure, and $\widetilde{g}(w)$ is $g(w)$ plus an additional perturbation term. This additional term is employed to ensure that the strongly convex parameter $\lambda + \kappa$ of $\widetilde{g}(w)$ satisfies $\lambda + \kappa \geq \frac{4}{n\gamma}$. Note that a smaller κ is preferable, as a larger κ leads to a severe bias between $f(w, \alpha)$ and $\widetilde{f}_s(w, \alpha)$.

Therefore, in the implementation of our ASPDC algorithms we simply use the smallest κ: $\kappa = \frac{4}{n\gamma} - \lambda$.

These calls of ASPDC produce a sequence $\widetilde{w}^s$, $s = 1, 2, \ldots$, which are the solutions of the corresponding approximate problem in Equation (12). Here, we need to prove that each running procedure of ASPDC from these calls can stop itself after finite epochs as well as that the output $\widetilde{w}^S$ satisfies the condition $|P(\widetilde{w}^S) - P(w^*)| \leq \epsilon$. In this condition, the variable w^* is the theoretical optimal solution of $P(w)$. These facts are illustrated in the following Theorem 2.

Theorem 2. *Algorithm 2 needs* $S \geq 1 + \frac{2}{\eta} log(\xi_1 \frac{1}{\epsilon})$ *epochs to approach the approximate solution* w^*, *where* $|P(\widetilde{w}^S) - P(w^*)| \leq \epsilon$.

The proof can be found in the Appendix A. The settings of the hyper parameters of Algorithm 2 are presented in the proof.

Algorithm 2 ASPDC-i

Parameter: $\lambda \leq \frac{4}{n\gamma}$, $\kappa = \frac{4}{n\gamma} - \lambda$, $\eta = \frac{\lambda}{\lambda + 2\kappa}$,
$\xi_1 = (1 + \eta^{-1})(P(\widetilde{w}^1) - D(\widetilde{\alpha}^1))$
Initialize: $\widetilde{w}^1 = 0$, $\widetilde{\alpha}^1 = 0$
for s= 1,2,3,... **do**
  $(\widetilde{w}^{s+1}, \widetilde{\alpha}^{s+1}, \epsilon_{s+1})$=ASPDC$(\widetilde{f}_s(w, \alpha), \widetilde{\alpha}^s, \frac{\eta}{2(1+\eta^{-1})}\xi_s)$
  $\xi_{s+1} = (1 - 0.5\eta)\xi_s$
end for
stop condition: $S \geq 1 + \frac{2}{\eta} log(\xi_1 \frac{1}{\epsilon})$

Output $\widetilde{w}^S$, $\widetilde{\alpha}^S$

To make for a fair comparison with other algorithms, we provide an realistic implementation version of Algorithm 2. This implementation version is shown in Algorithm 3. Here, the number of iterations in Algorithm 3 is set to be a constant m (e.g., $m = 2n$). As be demonstrated in the experiment section, this approach works well.

Algorithm 3 Implemented version of ASPDC-i

Parameter: $\lambda \leq \frac{4}{n\gamma}$, $\kappa = \frac{4}{n\gamma} - \lambda$
Initialize: $\widetilde{w}^0 = 0$, $\widetilde{\alpha}^0 = 0$
for $s = 1, 2, 3, \ldots, S$ **do**
  $\alpha^{(0)} = \widetilde{\alpha}^{(s-1)}$, $w^{(0)} = \nabla \widetilde{g}^*(-\frac{1}{n}\sum_{i=1}^n x_i \alpha_i^{(0)})$
  for $t = 0, 1, 2, \ldots, m-1$ **do**
    pick $i \in \{1, 2, \ldots, n\}$ under uniform distribute
    $\Delta\alpha_i^* = \arg\max_{\Delta\alpha_i \in \mathbb{R}} (\Delta\alpha_i x_i^T w^{(t)} - \phi_i^*(\alpha_i^{(t)} + \Delta\alpha_i))$
    $\alpha^{(t+1)} = \alpha^{(t)} + \Delta\alpha_i^* e_i$
    $w^{(t+1)} = \nabla \widetilde{g}^*(-\frac{1}{n}\sum_{i=1}^n x_i \alpha_i^{(t+1)})$
  end for
  $\widetilde{w}^s = w^{(m)}$, $\widetilde{\alpha}^s = \alpha^{(m)}$
end for

Output $\widetilde{w}^S$, $\widetilde{\alpha}^S$

6. Experiments

In this section, we evaluate the performance of our ASPDC algorithms along with several state-of-art algorithms for solving machine learning problems such as SVM. All the algorithms were implemented in C++ and executed through a Matlab interface. The experiments were performed on a PC with an Intel i5-4690 CPU and 16.0 GB RAM. The

source code and the detailed proofs can be downloaded from the GitHub website (https://github.com/lianghb6/ASPDC, access on 28 June 2022) and the datasets can be obtained from the LIBSVM website (https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, access on 28 June 2022).

As the computation processes of the problems are similar, in these experiments we mainly evaluated the practical performance of ASPDC for solving the following SVM optimization problem:

$$\min_{w\in\mathbb{R}^d}\{P(w)=\frac{1}{n}\sum_{i=1}^{n}\phi_i(w^Tx_i)+\frac{\lambda}{2}||w||_2^2\}$$

where ϕ_i is a smooth hinge loss, and is used in [1,5] as well.

$$\phi_i(w^Tx_i)=\begin{cases}0 & y_iw^Tx_i\geq 1\\ \frac{1}{2}-y_iw^Tx_i & y_iw^Tx_i\leq 0\\ \frac{1}{2}(1-y_iw^Tx_i)^2 & otherwise.\end{cases}$$

The corresponding convex–concave saddle point problem is as follows:

$$\min_{w\in\mathbb{R}^d}\max_{\alpha\in\mathbb{R}^n}\{f(w,\alpha)=\frac{1}{n}\sum_{i=1}^{n}[\alpha_iw^Tx_i-\phi_i^*(\alpha_i)]+\frac{\lambda}{2}||w||_2^2\}$$

where

$$\phi_i^*(\alpha_i)=\begin{cases}y_i\alpha_i+\frac{1}{2}\alpha_i^2 & -1\leq y_i\alpha_i\leq 0\\ +\infty & otherwise.\end{cases}$$

Under Assumption 3, the smooth parameter γ of ϕ_i is 1. The strongly convex parameter of $P(w)$ is λ, which comes from the regularized function $g(w)=\frac{\lambda}{2}||w||_2^2$.

In Figure 1 and Table 3, we show the cases when λ is relatively large (e.g., $10^{-2}, 10^{-3}, 10^{-4}$). We compare ASPDC (Algorithm 1) with state-of-art dual methods: the stochastic dual coordinate ascent method (SDCA)[5] and stochastic primal–dual coordinate method (SPDC) [1]. Note that accelerated stochastic dual ascent (ASDCA) [13] cannot be applied to this scenario, as ASDCA requires λ to be extremely small (i.e., $\lambda\leq\frac{1}{10n\gamma}$). We omit the comparison between ASPDC and the stochastic gradient descent method and its variants (e.g., SVRG [19] and Katyusha [26]), as there have already been extensive experiments using SPDC and this situation performed in the literature.

The horizontal axis in Figure 1 is the number of passes through the entire dataset, and the vertical axis is the logarithmic dual-gap. It can be seen from Figure 1 that ASPDC and SDCA have comparable performances on relatively large λ. With the same epoch, the dual-gap of ASPDC is lower than that of SPDC by two orders of magnitude after several epochs.

Figure 1 shows that both SDCA and ASPDC are faster than SPDC. This is because λ in Figure 1 is relative large (e.g., 0.01). In this case, the condition number of problems is relatively small. When the condition number is large, ASPDC and SPDC perform better than SDCA. In total, ASPDC is faster and is well suited for ill-conditioned problems.

Table 3. The running time for dual-gap approaches to the given precision (10^{-6}) when $\lambda=0.01$.

	SDCA	**SPDC**	**ASPDC**
a9a	0.505 s	1.311 s	0.636 s
ijcnn	0.984 s	1.438 s	1.183 s
covtype	11.502 s	20.526 s	14.972 s

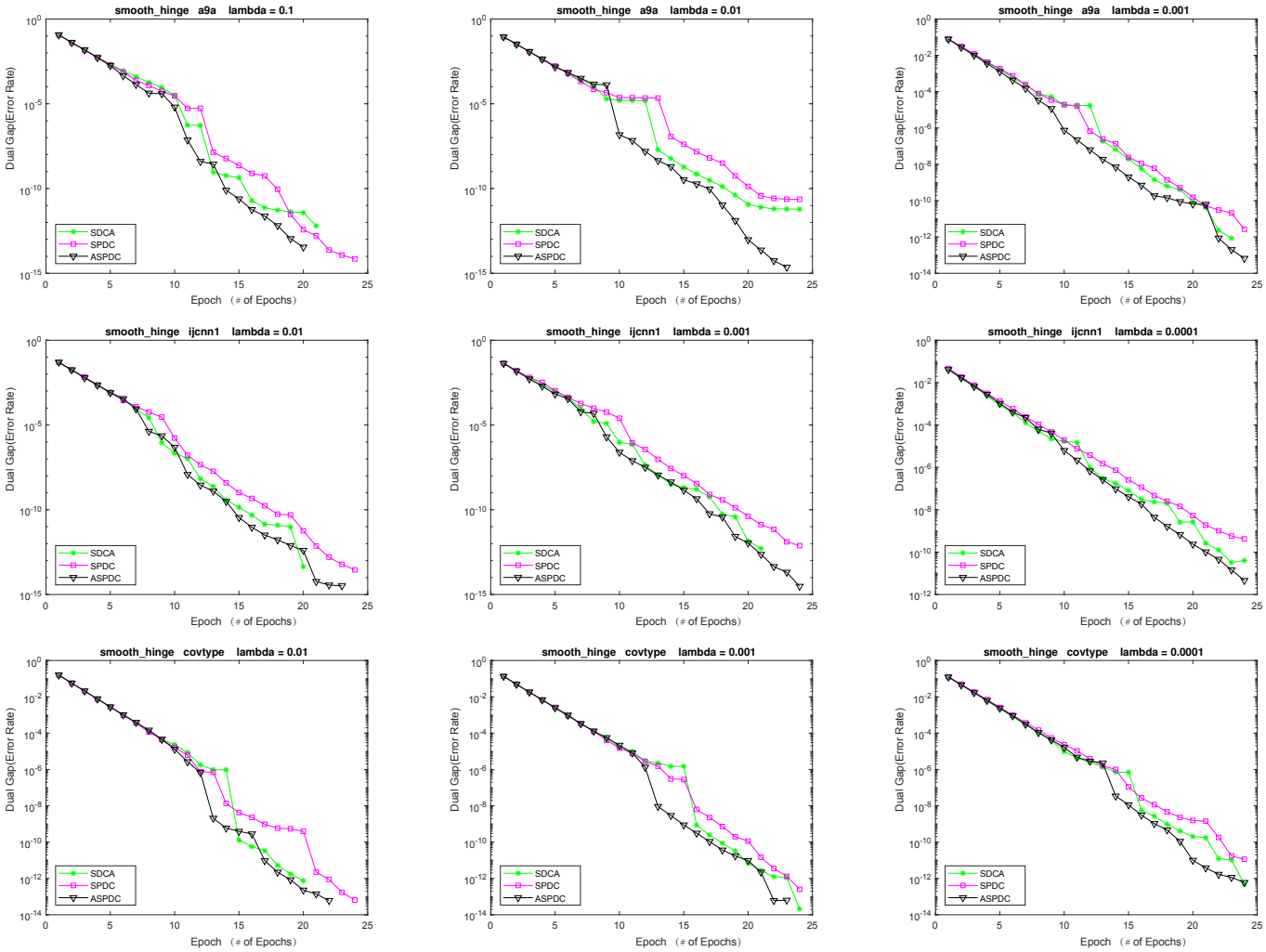

Figure 1. Dual-gap (y-axis) vs, the number of epochs (x-axis). Comparing ASPDC with other methods for smooth hinge SVM on real-world datasets with regularization coefficient $\lambda \in \{0.1, 0.01, 0.001, 0.0001\}$. The horizontal axis is the number of passes through the entire dataset, and the vertical axis is the logarithmic dual-gap.

Table 3 lists the needed running time for the dual-gaps of different algorithms to decrease to the given precision (e.g., $dual\ gap \leq 10^{-6}$) for different algorithms and datasets. Table 3 demonstrates that ASPDC and SDCA need less time to approach the given precision, and verifies that the convergence of ASPDC and SDCA is faster than SPDC. Table 4 presents the total running time for the algorithms to go through the entire dataset once to measure the per-iteration computation complexity. An algorithm with a shorter running time indicates that the algorithm has a lower per-iteration computation complexity. Table 4 shows that ASPDC and SDCA have a lower per-iteration complexity than SPDC. Among all of the running time results, ASPDC demonstrates both fast convergence and low per-iteration complexity when λ is large.

Table 4. The average running time for the algorithms to pass through the entire dataset once when $\lambda = 0.01$.

	SDCA	SPDC	ASPDC
a9a	0.029 s	0.052 s	0.028 s
ijcnn	0.053 s	0.061 s	0.052 s
covtype	0.650 s	0.840 s	0.644 s

We then tested the case when λ is relatively small (e.g., $\lambda \leq \frac{4}{\gamma n}$) and compared ASPDC-i with SDCA, SPDC, and ASDCA. Figure 2 plots the convergence results. Figure 2 shows that the convergences of SDCA, ASDCA, and SPDC are significantly slower than those of the same algorithms in Figure 1. The reason for this is that the condition number of the problem in this test case is larger than that in Figure 1. ASPDC-i performs much better in this experiment, as can be seen from Figure 2. ASPDC-i needs far fewer epochs than other algorithms to approach the same level of dual-gap. Additionally, ASPDC can approach a significantly lower dual-gap than the others with the same epochs.

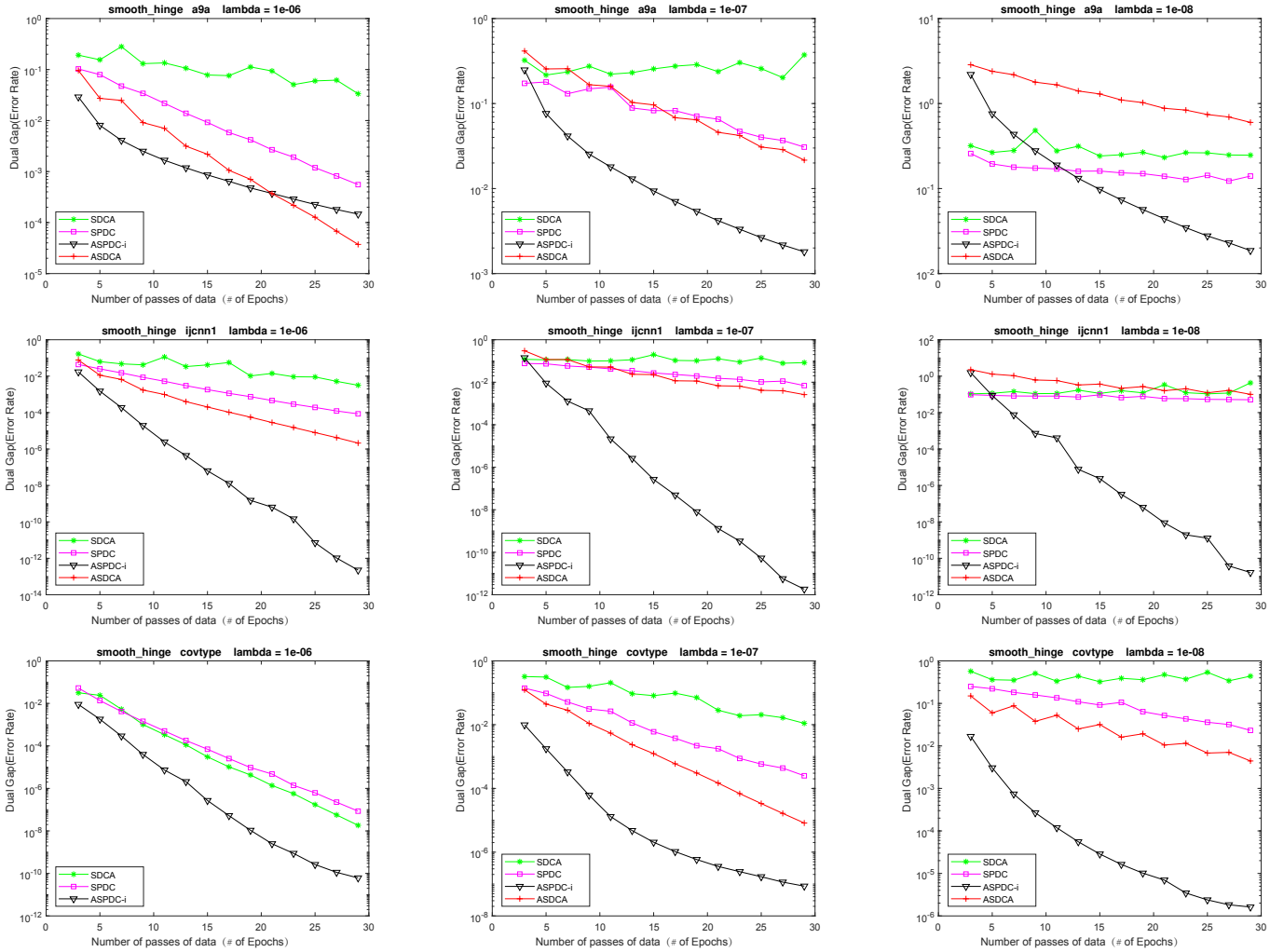

Figure 2. Dual-gap (y-axis) vs. the number of epochs (x-axis). Comparing ASPDC-i with other methods for smooth hinge SVM on real-world datasets with regularization coefficient $\lambda \in \{10^{-6}, 10^{-7}, 10^{-8}\}$. The horizontal axis is the number of passes through the entire dataset, and the vertical axis is the logarithmic dual-gap.

In addition, we compared ASPDC-i to a widely used non-dual-based algorithm, SVRG [19]. As SVRG is not dual-based, we directly compared its reduction speed of the primal value with ASPDC-i. Figure 3 shows that the convergence speed of ASPDC-i is faster than SVRG.

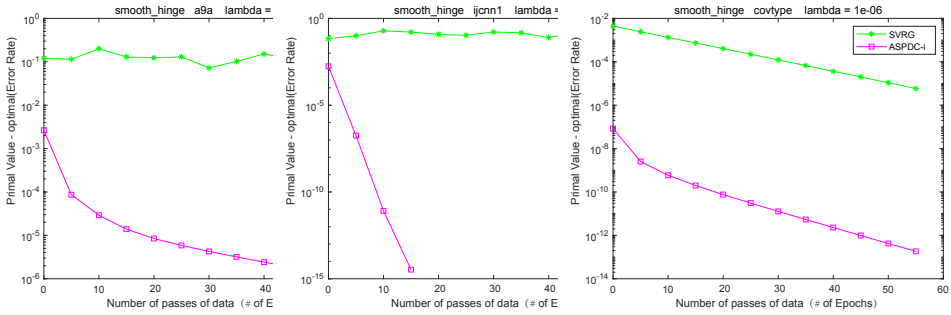

Figure 3. Optimal primal value (y-axis) vs. the number of epochs (x-axis): Comparing ASPDC-i with SVRG for smooth hinge SVM on real-world datasets with the regularization coefficient 10^{-6}. The x-axis is the number of passes through the entire dataset, and the y-axis is the logarithmic dual-gap.

Note that ASDCA cannot be applied to cases in which the dataset is covtype and $\lambda = 10^{-6}$, as ASDCA needs the extra condition $\lambda \leq \frac{1}{10n\gamma}$. Table 5 illustrates the running time that different algorithms spend to decrease the dual-gap to the given precision (e.g., 10^{-4}). Table 6 demonstrates the total running time for the algorithms to go through the entire dataset once. It shows that ASPDC and ASDCA have lower per-iteration complexity than SPDC. Although SDCA has low per-iteration complexity, its convergence is the slowest among these methods when λ is relatively small. We did not list the corresponding results of SDCA in Tables 5 and 6. In summary, the above experiments show that our proposed methods achieve both fast convergence and low per-iteration complexity.

Table 5. The running time for dual-gaps to approach the given precision (10^{-4}) when $\lambda = 10^{-6}$.

	ASDCA	**SPDC**	**ASPDC-i**
a9a	0.582 s	2.262 s	0.8464 s
ijcnn	0.994 s	3.127 s	2.033 s
covtype	8.407 s	91.132 s	47.734 s

Table 6. The average running time for the algorithms to pass through the entire dataset once when $\lambda = 10^{-6}$.

	ASDCA	**SPDC**	**ASPDC-i**
a9a	0.0165 s	0.0857 s	0.0167 s
ijcnn	0.0305 s	0.0821 s	0.0302 s
covtype	0.208 s	1.253 s	0.408 s

7. Conclusions and Future Work

In this paper, we propose two stochastic primal–dual coordinate methods, ASPDC and its accelerated variant version, ASPDC-i. These two algorithms are designed for the regularized empirical risk minimization problem. We proved the theoretical convergence guarantee of the algorithms and performed a series of experiments. The results illustrate that our methods achieve a good balance between low per-iteration computation complexity and fast convergence. The new convergence proof presented here uses Nesterov's estimation sequence technique and $g(w) = \frac{\lambda}{2}||w||_2^2$. We believe that it is possible to extend this proof to the more general regularized function $g(w)$; however, we leave this as a possibility for future work.

Author Contributions: Writing—original draft, H.L.; Data curation, F.S. and X.L.; Writing—review & editing, H.C., H.W. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Program of Guangzhou, China (No. 202002020045) and by the Meizhou Major Scientific and Technological Innovation Platforms and Projects of Guangdong Provincial Science & Technology Plan Projects under Grant No. 2019A0102005.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1. Proof of Lemma 1

We prove the following equations: $P(w) = \max_{\alpha \in \mathbb{R}^n} f(w, \alpha)$, $D(\alpha) = \min_{w \in \mathbb{R}^d} f(w, \alpha)$ and $P(w^*) = D(\alpha^*) = f(w^*, \alpha^*)$. We first prove $P(w) = \max_{\alpha \in \mathbb{R}^n} f(w, \alpha)$.

Proof.

$$\begin{aligned}
&\max_{\alpha \in \mathbb{R}^n} f(w, \alpha) \\
&= \max_{\alpha \in \mathbb{R}^n} \{\tfrac{1}{n} \textstyle\sum_{i=1}^n [\alpha_i w^T x_i - \phi_i^*(\alpha_i)] + g(w)\} \\
&= \max_{\alpha \in \mathbb{R}^n} \{\tfrac{1}{n} \textstyle\sum_{i=1}^n [\alpha_i w^T x_i - \phi_i^*(\alpha_i)]\} + g(w) \\
&= \tfrac{1}{n} \textstyle\sum_{i=1}^n \max_{\alpha_i \in \mathbb{R}} \{\alpha_i w^T x_i - \phi_i^*(\alpha_i)\} + g(w) \\
&= \tfrac{1}{n} \textstyle\sum_{i=1}^n \phi_i(w^T x_i) + g(w) \\
&= P(w)
\end{aligned} \tag{A1}$$

In the last equation, we use the Conjugate Theorem (Convex Optimization Theory). Then, we prove that $D(\alpha) = \min_{w \in \mathbb{R}^d} f(w, \alpha)$.

$$\begin{aligned}
&\min_{w \in \mathbb{R}^d} f(w, \alpha) \\
&= \min_{w \in \mathbb{R}^d} \{\frac{1}{n} \sum_{i=1}^n [\alpha_i w^T x_i - \phi_i^*(\alpha_i)] + g(w)\} \\
&= \frac{-1}{n} \sum_{i=1}^n \phi_i^*(\alpha_i) + \min_{w \in \mathbb{R}^d} \{(\frac{1}{n} \sum_{i=1}^n \alpha_i x_i)^T w + g(w)\} \\
&= \frac{-1}{n} \sum_{i=1}^n \phi_i^*(\alpha_i) - \max_{w \in \mathbb{R}^d} \{(\frac{-1}{n} \sum_{i=1}^n \alpha_i x_i)^T w - g(w)\} \\
&= \frac{-1}{n} \sum_{i=1}^n \phi_i^*(\alpha_i) - g^*(\frac{-1}{n} \sum_{i=1}^n \alpha_i x_i) \\
&= D(\alpha)
\end{aligned}$$

The proof of $P(w^*) = D(\alpha^*) = f(w^*, \alpha^*)$ can be found in [1]. □

Appendix A.2. Proof of Lemma 2

Proof. When $g(w) = \frac{\lambda}{2}||w||_2^2$, the primal objective can be written as follows:

$$P(w) = \frac{1}{n} \sum_{i=1}^n \phi_i(w^T x_i) + \frac{\lambda}{2}||w||_2^2. \tag{A2}$$

The corresponding dual objective is

$$D(\alpha) = \frac{-1}{n} \sum_{i=1}^n \phi_i^*(\alpha_i) - \frac{\lambda}{2}||\frac{-1}{\lambda n} \sum_{i=1}^n \alpha_i x_i||_2^2. \tag{A3}$$

Note that at through the algorithm, we can set

$$w^{(t)} = \frac{-1}{\lambda n} \sum_{i=1}^{n} \alpha_i^{(t)} x_i. \tag{A4}$$

Thus, the $D(\alpha^{(t)})$ can be written as

$$D(\alpha^{(t)}) = \frac{-1}{n} \sum_{i=1}^{n} \phi_i^*(\alpha_i^{(t)}) - \frac{\lambda}{2} ||w^{(t)}||_2^2. \tag{A5}$$

Suppose we have $\alpha^{(t)}$ and that the $i-th$ coordinate is chosen at iteration $t+1$:

$$\begin{aligned} & D(\alpha^{(t+1)}) - D(\alpha^{(t)}) \\ & = \underbrace{-\frac{1}{n}\phi_i^*(\alpha_i^{(t+1)}) - \frac{\lambda}{2}||w^{(t)} - \frac{1}{\lambda n}\Delta\alpha_i^* x_i||_2^2}_{R_1} \\ & - \underbrace{\{-\frac{1}{n}\phi_i^*(\alpha_i^{(t)}) - \frac{\lambda}{2}||w^{(t)}||_2^2\}}_{R_2}. \end{aligned} \tag{A6}$$

The variables in the algorithm are as follows:

$$\begin{aligned} \Delta\alpha_i^* &= \arg\max_{d\in\mathbb{R}}(dx_i^T w^{(t)} - \phi_i^*(\alpha_i^{(t)} + d)) \\ &= \arg\max_{d\in\mathbb{R}}((\alpha_i^{(t)} + d)x_i^T w^{(t)} - \phi_i^*(\alpha_i^{(t)} + d)) \\ &= \arg\max_{\alpha_i^{(t)}+d\in\mathbb{R}}((\alpha_i^{(t)} + d)x_i^T w^{(t)} - \phi_i^*(\alpha_i^{(t)} + d)) \\ &= \arg\max_{\beta\in\mathbb{R}}(\beta x_i^T w^{(t)} - \phi_i^*(\beta)) - \alpha_i^{(t)}, \end{aligned} \tag{A7}$$

where in the last inequality we define $\beta = \alpha_i^{(t)} + d$, and correspondingly have $\beta^* = \alpha_i^{(t)} + \Delta\alpha_i^*$.

$$
\begin{aligned}
&R_1 = -\tfrac{1}{n}\phi_i^*(\alpha_i^{(t+1)}) - \tfrac{\lambda}{2}||w^{(t)} - \tfrac{1}{\lambda n}\Delta\alpha_i^* x_i||_2^2 \\
&= -\tfrac{1}{n}\phi_i^*(\alpha_i^{(t)} + \Delta\alpha_i^*) + \tfrac{1}{n}\Delta\alpha_i^* x_i^T w^{(t)} \\
&-\tfrac{1}{2\lambda n^2}||x_i||_2^2(\Delta\alpha_i^*)^2 - \tfrac{\lambda}{2}||w^{(t)}||_2^2 \\
&= \tfrac{1}{n}\{\max_{d\in\mathbb{R}}(dx_i^T w^{(t)} - \phi_i^*(\alpha_i^{(t)} + d))\} \\
&-\tfrac{1}{2\lambda n^2}||x_i||_2^2(\Delta\alpha_i^*)^2 - \tfrac{\lambda}{2}||w^{(t)}||_2^2 \\
&\overset{①}{\geq} \tfrac{1}{n}\{q(\beta^* - \alpha_i^{(t)})x_i^T w^{(t)} - \phi_i^*(\alpha_i^{(t)} + q(\beta^* - \alpha_i^{(t)}))\} \\
&-\tfrac{1}{2\lambda n^2}||x_i||_2^2(\Delta\alpha_i^*)^2 - \tfrac{\lambda}{2}||w^{(t)}||_2^2 \\
&= -\tfrac{1}{n}\phi_i^*((1-q)\alpha_i^{(t)} + q\beta^*) + \tfrac{1}{n}q(\beta^* - \alpha_i^{(t)})x_i^T w^{(t)} \\
&-\tfrac{1}{2\lambda n^2}||x_i||_2^2(\Delta\alpha_i^*)^2 - \tfrac{\lambda}{2}||w^{(t)}||_2^2 \\
&\overset{②}{\geq} -\tfrac{1}{n}\{q\phi_i^*(\beta^*) + (1-q)\phi_i^*(\alpha_i^{(t)}) - \tfrac{\gamma q(1-q)}{2}(\beta^* - \alpha_i^{(t)})^2\} \\
&+\tfrac{1}{n}q(\beta^* - \alpha_i^{(t)})x_i^T w^{(t)} - \tfrac{1}{2\lambda n^2}||x_i||_2^2(\Delta\alpha_i^*)^2 - \tfrac{\lambda}{2}||w^{(t)}||_2^2 \\
&\geq \tfrac{q}{n}\{-\phi_i^*(\beta^*) + \beta^* x_i^T w^{(t)}\} - \tfrac{1-q}{n}\phi_i^*(\alpha_i^{(t)}) \\
&+\tfrac{\gamma(1-q)q}{2n}(\beta^* - \alpha_i^{(t)})^2 - \tfrac{q}{n}\alpha_i^{(t)} x_i^T w^{(t)} \\
&-\tfrac{1}{2\lambda n^2}||x_i||_2^2(\Delta\alpha_i^*)^2 - \tfrac{\lambda}{2}||w^{(t)}||_2^2
\end{aligned}
\tag{A8}
$$

where $q \in (0,1)$ in the inequality ①, while in the inequality ② we use the fact that if ϕ_i is $\frac{1}{\gamma}$ smooth, then ϕ_i^* is γ strong convex.

On the one hand, according to (A7), we obtain

$$\beta^* = \arg\max_{\beta\in\mathbb{R}}(\beta x_i^T w^{(t)} - \phi_i^*(\beta)). \tag{A9}$$

This implies that

$$x_i^T w^{(t)} = \nabla\phi_i^*(\beta^*). \tag{A10}$$

On the other hand, by the definition of the convex conjugate function, we have $\phi_i^{**}(x_i^T w^{(t)}) = \max_{\beta\in\mathbb{R}}(\beta x_i^T w^{(t)} - \phi_i^*(\beta))$. According to the Fenchel conjugate sub-gradient theorem, we have

$$
\begin{aligned}
&x_i^T w^{(t)} = \nabla\phi_i^*(\beta^*) \Longleftrightarrow \beta^* x_i^T w^{(t)} - \phi_i^*(\beta^*) \\
&= \phi_i^{**}(x_i^T w^{(t)}) \overset{③}{=} \phi_i(x_i^T w^{(t)}),
\end{aligned}
\tag{A11}
$$

where in ③ we apply the Fenchel Dual theorem.

Combined with (A8) and (A11), we obtain

$$
\begin{aligned}
&R_1 \geq \tfrac{q}{n}\{\phi_i(x_i^T w^{(t)}) + \phi_i^*(\alpha_i^{(t)}) - \alpha_i^{(t)} x_i^T w^{(t)}\} \\
&+\tfrac{\gamma(1-q)q}{2n}(\beta^* - \alpha_i^{(t)})^2 - \tfrac{1}{2\lambda n^2}||x_i||_2^2(\Delta\alpha_i^*)^2 \\
&+\{\underbrace{-\frac{1}{n}\phi_i^*(\alpha_i^{(t)}) - \frac{\lambda}{2}||w^{(t)}||_2^2}_{R_2}\}.
\end{aligned}
\tag{A12}
$$

Combining $\beta^* = \alpha_i^{(t)} + \Delta\alpha_i^*$ with (A6) and (A12), we have

$$\begin{aligned} & D(\alpha^{(t+1)}) - D(\alpha^{(t)}) \\ & \geq \tfrac{q}{n}\{\phi_i(x_i^T w^{(t)}) + \phi_i^*(\alpha_i^{(t)}) - \alpha_i^{(t)} x_i^T w^{(t)}\} \\ & + \{\tfrac{\gamma(1-q)q}{2n} - \tfrac{1}{2(\lambda)n^2}||x_i||_2^2\}(\Delta\alpha_i^*)^2 \\ & \geq \tfrac{q}{n}\{\phi_i(x_i^T w^{(t)}) + \phi_i^*(\alpha_i^{(t)}) - \alpha_i^{(t)} x_i^T w^{(t)}\} \\ & + \{\tfrac{\gamma(1-q)q}{2n} - \tfrac{1}{2(\lambda n^2}\}(\Delta\alpha_i^*)^2, \end{aligned} \tag{A13}$$

where in the last inequality we use the assumption $||x_i||_2^2 \leq 1$. Note that we have supposed that the $i-th$ coordinate of α is chosen, thus, we use the expectation of (A13) with respect to i, obtaining

$$\begin{aligned} & E\{D(\alpha^{(t+1)}) - D(\alpha^{(t)})\} \\ & \geq \tfrac{q}{n}\tfrac{1}{n}\textstyle\sum_{i=1}^n\{\phi_i(x_i^T w^{(t)}) + \phi_i^*(\alpha_i^{(t)}) - \alpha_i^{(t)} x_i^T w^{(t)}\} \\ & + \{\tfrac{\gamma(1-q)q}{2n} - \tfrac{1}{2\lambda n^2}\}\tfrac{1}{n}\textstyle\sum_{i=1}^n(\Delta\alpha_i^*)^2. \end{aligned} \tag{A14}$$

Recall that

$$\begin{aligned} & P(w^{(t)}) - D(\alpha^{(t)}) \\ & = \tfrac{1}{n}\textstyle\sum_{i=1}^n\{\phi_i(x_i^T w^{(t)}) + \phi_i^*(\alpha_i^{(t)})\} + \lambda||w^{(t)}||_2^2 \\ & \overset{④}{=} \tfrac{1}{n}\textstyle\sum_{i=1}^n\{\phi_i(x_i^T w^{(t)}) + \phi_i^*(\alpha_i^{(t)}) - \alpha_i^{(t)} x_i^T w^{(t)}\}, \end{aligned} \tag{A15}$$

where in ④ we use the fact that $w^{(t)} = \frac{-1}{\lambda n}\sum_{i=1}^n \alpha_i^{(t)} x_i$.

Combined (A14) with (A15), we obtain

$$\begin{aligned} & E\{D(\alpha^{(t+1)}) - D(\alpha^{(t)})\} \\ & \geq \tfrac{q}{n}\{P(w^{(t)}) - D(\alpha^{(t)})\} + \{\tfrac{\gamma(1-q)q}{2n} - \tfrac{1}{2\lambda n^2}\}\tfrac{1}{n}\textstyle\sum_{i=1}^n(\Delta\alpha_i^*)^2. \end{aligned} \tag{A16}$$

Using $q = 1/2$ and $\lambda \geq \frac{4}{n\gamma}$, we have $\frac{\gamma(1-q)q}{2n} - \frac{1}{2\lambda n^2} \geq 0$, and

$$E\{D(\alpha^{(t+1)}) - D(\alpha^{(t)})\} \geq \frac{1}{2n}\{P(w^{(t)}) - D(\alpha^{(t)})\}. \tag{A17}$$

Note that $\alpha^* = \arg\max_\alpha D(\alpha)$; it is well known that $P(w^{(t)}) \geq D(\alpha^*) \geq D(\alpha^{(t)})$. Combined with (A17), we obtain

$$\begin{aligned} & \tfrac{1}{2n}\{D(\alpha^*) - D(\alpha^{(t)})\} \\ & \leq \tfrac{1}{2n}\{P(w^{(t)}) - D(\alpha^{(t)})\} \\ & \leq E\{D(\alpha^{(t+1)}) - D(\alpha^{(t)})\} \\ & = E\{D(\alpha^{(t+1)}) - D(\alpha^*) + D(\alpha^*) - D(\alpha^{(t)})\} \\ & = \{D(\alpha^*) - D(\alpha^{(t)})\} - E\{D(\alpha^*) - D(\alpha^{(t+1)})\}. \end{aligned} \tag{A18}$$

This further implies that

$$E\{D(\alpha^*) - D(\alpha^{(t+1)})\} \leq (1 - \frac{1}{2n})\{D(\alpha^*) - D(\alpha^{(t)}). \tag{A19}$$

Until now, we have assumed that $\alpha^{(t)}$ is known and the expectation is for random variable i; if below we take this expectation with all the history i, we obtain

$$E\{D(\alpha^*) - D(\alpha^{(t+1)})\} \leq (1 - \frac{1}{2n})^{(t+1)}\{D(\alpha^*) - D(\alpha^{(0)}). \tag{A20}$$

In addition, it can be known from (A17) that

$$\begin{aligned}
&\frac{1}{2n}E\{P(w^{(t)}) - D(\alpha^{(t)})\} \leq E\{D(\alpha^{(t+1)}) - D(\alpha^{(t)})\} \\
&= \{D(\alpha^*) - D(\alpha^{(t)})\} - E\{D(\alpha^*) - D(\alpha^{(t+1)})\} \\
&\leq \{D(\alpha^*) - D(\alpha^{(t)})\} \\
&\leq (1 - \frac{1}{2n})^t\{D(\alpha^*) - D(\alpha^{(0)})\}
\end{aligned}$$

This implies that $E\{P(w^{(t)}) - D(\alpha^{(t)})\} \leq 2n(1 - \frac{1}{2n})^t\{D(\alpha^*) - D(\alpha^{(0)})\}$. □

References

1. Zhang, Y.; Xiao, L. Stochastic Primal-Dual Coordinate Method for Regularized Empirical Risk Minimization. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 353–361.
2. Ruppert, D. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. *Publ. Am. Stat. Assoc.* **2010**, *99*, 567–567.
3. Chiang, W.; Lee, M.; Lin, C. Parallel Dual Coordinate Descent Method for Large-scale Linear Classification in Multi-core Environments. In *KDD '16, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco , CA, USA, 13–17 August 2016*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1485–1494.
4. Hsieh, C.; Chang, K.; Lin, C.; Keerthi, S.S.; Sundararajan, S. A dual coordinate descent method for large-scale linear SVM. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 408–415.
5. Shalevshwartz, S.; Zhang, T. Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *J. Mach. Learn. Res.* **2012**, *14*, 2013.
6. Chang, K.W.; Hsieh, C.J.; Lin, C.J. Coordinate Descent Method for Large-scale L2-loss Linear Support Vector Machines. *J. Mach. Learn. Res.* **2008**, *9*, 1369–1398.
7. Platt, J.C. *Fast Training of Support Vector Machines Using Sequential Minimal Optimization*; MIT Press: Cambridge, MA, USA, 1999; pp. 185–208.
8. Naskovska, K.; Lau, S.; Korobkov, A.A.; Haueisen, J.; Haardt, M. Coupled CP decomposition of simultaneous MEG-EEG signals for differentiating oscillators during photic driving. *Front. Neurosci.* **2020**, *14*, 261.
9. Lee, S.; Kim, E.; Kim, C.; Kim, K. Localization with a mobile beacon based on geometric constraints in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 5801–5805.
10. Wang, J.; Dong, P.; Jing, Z.; Cheng, J. Consensus-based filter for distributed sensor networks with colored measurement noise. *Sensors* **2018**, *18*, 3678.
11. Anastassiu, H.T.; Vougioukas, S.; Fronimos, T.; Regen, C.; Petrou, L.; Zude, M.; Käthner, J. A computational model for path loss in wireless sensor networks in orchard environments. *Sensors* **2014**, *14*, 5118–5135.
12. Deng, X.; Yin, L.; Peng, S.; Ding, M. An iterative algorithm for solving ill-conditioned linear least squares problems. *Geod. Geodyn.* **2015**, *6*, 453–459. https://doi.org/10.1016/j.geog.2015.06.004.
13. Shalevshwartz, S.; Zhang, T. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In Proceedings of the International Conference on Machine Learning, Bejing, China , 21–26 June 2014.
14. Bauschke, H.H.; Combettes, P.L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*; Springer: New York, NY, USA, 2011; Volume 408.
15. Güler, O. New proximal point algorithms for convex minimization. *SIAM J. Optim.* **1992**, *2*, 649–664.
16. Nesterov, Y. *Introductory Lectures on Convex Optimization*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2014; pp. xviii, 236.
17. Frostig, R.; Ge, R.; Kakade, S.; Sidford, A. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2540–2548.
18. Lin, H.; Mairal, J.; Harchaoui, Z. A Universal Catalyst for First-Order Optimization. Available online: https://proceedings.neurips.cc/paper/2015/hash/c164bbc9d6c72a52c599bbb43d8db8e1-Abstract.html (accessed on 29 June 2022).
19. Johnson, R.; Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In Proceedings of the Advances in Neural Information Processing Systems, Tahoe, CA, USA, 5–10 December 2013; pp. 315–323.
20. Xiao, L.; Zhang, T. A proximal stochastic gradient method with progressive variance reduction. *SIAM J. Optim.* **2014**, *24*, 2057–2075.
21. Zhang, Y.; Xiao, L. Stochastic primal-dual coordinate method for regularized empirical risk minimization. *J. Mach. Learn. Res.* **2017**, *18*, 2939–2980.
22. Devolder, O.; Glineur, F.; Nesterov, Y. First-order methods of smooth convex optimization with inexact oracle. *Math. Program.* **2014**, *146*, 37–75.
23. Schmidt, M.; Roux, N.L.; Bach, F.R. Convergence rates of inexact proximal-gradient methods for convex optimization. In Proceedings of the Advances in Neural Information Processing Systems, Granada, Spain, 12–15 December 2011; pp. 1458–1466.
24. Hiriart-Urruty, J.B.; Lemaréchal, C. *Fundamentals of Convex Analysis*; Springer Science & Business Media: New York, NY, USA, 2012.

25. Bertsekas, D.P. *Convex Optimization Theory*; Athena Scientific Belmont: Belmont, MA, USA, 2009.
26. Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. In Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, Montreal, QC, Canada, 19–23 June 2017; pp. 1200–1205.