Math. Program., Ser. A (2016) 155:105–145
DOI 10.1007/s10107-014-0839-0

FULL LENGTH PAPER

Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization

Shai Shalev-Shwartz · Tong Zhang

Received: 22 December 2013 / Accepted: 31 October 2014 / Published online: 20 November 2014

Abstract We introduce a proximal version of the stochastic dual coordinate ascent method and show how to accelerate the method using an inner-outer iteration procedure. We analyze the runtime of the framework and obtain rates that improve state-of-the-art results for various key machine learning optimization problems including SVM, logistic regression, ridge regression, Lasso, and multiclass SVM. Experiments validate our theoretical findings.

Mathematics Subject Classification 90C06 · 90C15 · 90C25

1 Introduction

Many machine learning algorithms fall into the paradigm of *regularized loss minimization* with respect to convex loss functions (see for example [23, Chapter 13]). In this paper we describe and analyze a new optimization algorithm for solving the regularized loss minimization optimization problem.

Formally, consider a sequence of n matrices $X_1, \ldots, X_n$ in $\mathbb{R}^{d \times k}$ (referred to as instances), and a corresponding sequence $\phi_1, \ldots, \phi_n$ of vector convex functions

This paper replaces and improves an earlier arxiv version [25]. An extended abstract of the paper was presented in ICML 2014.

S. Shalev-Shwartz
School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel

T. Zhang (✉)
Department of Statistics, Rutgers University, Piscataway, NJ, USA
e-mail: tzhang@stat.rutgers.edu

T. Zhang
Baidu Inc., Beijing, China

defined on $\mathbb{R}^k$ (referred to as loss functions); moreover consider a convex function $g(\cdot)$ defined on $\mathbb{R}^d$ (referred to as a regularizer), and a real number $\lambda \geq 0$ (referred to as a regularization parameter). Our goal is to solve:

$$\min_{w \in \mathbb{R}^d} P(w) \quad \text{where} \quad P(w) = \left[\frac{1}{n} \sum_{i=1}^{n} \phi_i(X_i^\top w) + \lambda g(w) \right]. \tag{1}$$

For example, in ridge regression the regularizer is $g(w) = \frac{1}{2}\|w\|_2^2$, the instances are column vectors, and for every i the i'th loss function is $\phi_i(a) = \frac{1}{2}(a - y_i)^2$, for some scalar y_i.

Let $w^* = \operatorname{argmin}_w P(w)$ (we will later make assumptions that imply that w^* is unique). We say that w is ϵ-accurate if $P(w) - P(w^*) \leq \epsilon$. Our main result is a new algorithm for solving (1). If g is 1-strongly convex and each ϕ_i is $(1/\gamma)$-smooth (meaning that its gradient is $(1/\gamma)$-Lipschitz), then our algorithm finds, with probability of at least $1 - \delta$, an ϵ-accurate solution to (1) in time

$$\begin{aligned} &O\left(d\left(n + \min\left\{\frac{1}{\lambda\gamma}, \sqrt{\frac{n}{\lambda\gamma}}\right\}\right) \log(1/\epsilon)\ \log(1/\delta)\ \max\{1, \log^2(1/(\lambda\gamma n))\}\right) \\ &\quad = \tilde{O}\left(d\left(n + \min\left\{\frac{1}{\lambda\gamma}, \sqrt{\frac{n}{\lambda\gamma}}\right\}\right)\right). \end{aligned}$$

This applies, for example, to ridge regression and to logistic regression with L_2 regularization. The O notation hides constants terms and the $\tilde{O}$ notation hides constants and logarithmic terms. We make these explicit in the formal statement of our theorems.

Intuitively, we can think of $\frac{1}{\lambda\gamma}$ as the condition number of the problem. If the condition number is $O(n)$ then our runtime becomes $\tilde{O}(dn)$. This means that the runtime is nearly linear in the data size. This matches the recent result of Shalev-Shwartz and Zhang [26], Le Roux et al. [16], but our setting is significantly more general. When the condition number is much larger than n, our runtime becomes $\tilde{O}(d\sqrt{\frac{n}{\lambda\gamma}})$. This significantly improves over the result of [16,26]. It also significantly improves over the runtime of accelerated gradient descent due to Nesterov [19], which is $\tilde{O}(d\,n\sqrt{\frac{1}{\lambda\gamma}})$.

By applying a smoothing technique to ϕ_i, we also derive a method that finds an ϵ-accurate solution to (1) assuming that each ϕ_i is $O(1)$-Lipschitz, and obtain the runtime

$$\tilde{O}\left(d\left(n + \min\left\{\frac{1}{\lambda\epsilon}, \sqrt{\frac{n}{\lambda\epsilon}}\right\}\right)\right).$$

This applies, for example, to SVM with the hinge-loss. It significantly improves over the rate $\frac{d}{\lambda\epsilon}$ of SGD (e.g. [27]), when $\frac{1}{\lambda\epsilon} \gg n$.

We can also apply our results to non-strongly convex regularizers (such as the L_1 norm regularizer), or to non-regularized problems, by adding a slight L_2 regularization. For example, for L_1 regularized problems, and assuming that each ϕ_i is $(1/\gamma)$-smooth, we obtain the runtime of

$$\tilde{O}\left(d\left(n+\min\left\{\frac{1}{\epsilon\gamma}, \sqrt{\frac{n}{\epsilon\gamma}}\right\}\right)\right).$$

This applies, for example, to the Lasso problem, in which the goal is to minimize the squared loss plus an L_1 regularization term.

To put our results in context, in the table below we specify the runtime of various algorithms (with $k = 1$) for three key machine learning applications; SVM in which $\phi_i(a) = \max\{0, 1 - a\}$ and $g(w) = \frac{1}{2}\|w\|_2^2$, Lasso in which $\phi_i(a) = \frac{1}{2}(a - y_i)^2$ and $g(w) = \sigma\|w\|_1$, and Ridge Regression in which $\phi_i(a) = \frac{1}{2}(a - y_i)^2$ and $g(w) = \frac{1}{2}\|w\|_2^2$. Note that we are particularly interested in the dependency of different algorithms on $d, n, 1/\lambda, 1/\epsilon$ (for typical applications, λ is generally of the order $1/\sqrt{n}$ or smaller). To avoid cluttering and focus on these quantities, we ignore constants and logarithmic terms. We also assume, without loss of generality, that the instance vectors are normalized to have a unit norm (that is, $R = 1$ in Theorem 1). Additional applications, and a more detailed runtime comparison to previous work, are given in Sect. 5. In the table below, SGD stands for Stochastic Gradient Descent, and AGD stands for Accelerated Gradient Descent.

Problem	Algorithm	Runtime
SVM	SGD/SDCA [26,27,29]	$\frac{d}{\lambda\epsilon}$
	AGD [2,18,19]	$dn\sqrt{\frac{1}{\lambda\epsilon}}$
	This paper	$d\left(n+\min\{\frac{1}{\lambda\epsilon}, \sqrt{\frac{n}{\lambda\epsilon}}\}\right)$
Lasso	SGD and variants (e.g. [22,30,31])	$\frac{d}{\epsilon^2}$
	Stochastic Coordinate Descent [20,24]	$\frac{dn}{\epsilon}$
	FISTA [2,19]	$dn\sqrt{\frac{1}{\epsilon}}$
	This paper	$d\left(n+\min\{\frac{1}{\epsilon}, \sqrt{\frac{n}{\epsilon}}\}\right)$
Ridge Regression	Exact	$d^2n + d^3$
	SAG/SDCA [16,26]	$d\left(n+\frac{1}{\lambda}\right)$
	AGD [2,18,19]	$dn\sqrt{\frac{1}{\lambda}}$
	This paper	$d\left(n+\min\{\frac{1}{\lambda}, \sqrt{\frac{n}{\lambda}}\}\right)$

Technical contribution Our algorithm combines two ideas. The first is a proximal version of stochastic dual coordinate ascent (SDCA).[1] In particular, we generalize the recent analysis of [26] in two directions. First, we allow the regularizer, g, to be a general strongly convex function (and not necessarily the squared Euclidean norm). This allows us to consider non-smooth regularization function, such as the L_1 regularization. Second, we allow the loss functions, ϕ_i, to be vector valued functions which are smooth (or Lipschitz) with respect to a general norm. This generalization is useful in

[1] Technically speaking, it may be more accurate to use the term *randomized* dual coordinate ascent, instead of *stochastic* dual coordinate ascent. This is because our algorithm makes more than one pass over the data, and therefore cannot work directly on distributions with infinite support. However, following the convention in the prior machine learning literature, we do not make this distinction.

multiclass applications. As in [26], the runtime of this procedure is $\tilde{O}\left(d\left(n+\frac{1}{\lambda\gamma}\right)\right)$. This would be a nearly linear time (in the size of the data) if $\frac{1}{\lambda\gamma} = O(n)$. Our second idea deals with the case $\frac{1}{\lambda\gamma} \gg n$ by iteratively approximating the objective function P with objective functions that have a stronger regularization. In particular, each iteration of our acceleration procedure involves approximate minimization of $P(w) + \frac{\kappa}{2}\|w-y\|_2^2$, with respect to w, where y is a vector obtained from previous iterates and κ is order of $1/(\gamma n)$. The idea is that the addition of the relatively strong regularization makes the runtime of our proximal stochastic dual coordinate ascent procedure be $\tilde{O}(dn)$. And, with a proper choice of y at each iteration, we show that the sequence of solutions of the problems with the added regularization converge to the minimum of P after $\sqrt{\frac{1}{\lambda\gamma n}}$ iterations. This yields the overall runtime of $d\sqrt{\frac{n}{\lambda\gamma}}$.

Additional related work As mentioned before, our first contribution is a proximal version of the stochastic dual coordinate ascent method and extension of the analysis given in Shalev-Shwartz and Zhang [26]. Stochastic dual coordinate ascent has also been studied in Collins et al. [3] but in more restricted settings than the general problem considered in this paper. One can also apply the analysis of stochastic coordinate descent methods given in Richtárik and Takáč [20] (see also [17]) on the dual problem. However, here we are interested in understanding the primal sub-optimality, hence an analysis which only applies to the dual problem is not sufficient.

The generality of our approach allows us to apply it for multiclass prediction problems. We discuss this in detail later on in Sect. 5. Recently, [14] derived a stochastic coordinate ascent for structural SVM based on the Frank–Wolfe algorithm. Although with different motivations, for the special case of multiclass problems with the hinge-loss, their algorithm ends up to be the same as our proximal dual ascent algorithm (with the same rate). Our approach allows to accelerate the method and obtain an even faster rate.

The proof of our acceleration method adapts Nesterov's estimation sequence technique, studied in Devolder et al. [7], Schmidt et al. [21], to allow approximate and stochastic proximal mapping. See also [1,6]. In particular, it relies on similar ideas as in Proposition 4 of [21]. However, our specific requirement is different, and the proof presented here is different and significantly simpler than that of [21].

There have been several attempts to accelerate stochastic optimization algorithms. See for example [4,12,13] and the references therein. However, the runtime of these methods have a polynomial dependence on $1/\epsilon$ even if ϕ_i are smooth and g is λ-strongly convex, as opposed to the logarithmic dependence on $1/\epsilon$ obtained here. As in [16,26], we avoid the polynomial dependence on $1/\epsilon$ by allowing more than a single pass over the data.

2 Preliminaries

All the functions we consider in this paper are proper convex functions over a Euclidean space. We use $\mathbb{R}$ to denote the set of real numbers and to simplify our notation, when we use $\mathbb{R}$ to denote the range of a function f we in fact allow f to output the value $+\infty$.

Given a function $f : \mathbb{R}^d \to \mathbb{R}$ we denote its **conjugate** function by

$$f^*(y) = \sup_x \; [y^\top x - f(x)].$$

Given a norm $\|\cdot\|_P$ we denote the **dual norm** by $\|\cdot\|_D$ where

$$\|y\|_D = \sup_{x:\|x\|_P=1} y^\top x.$$

We use $\|\cdot\|$ or $\|\cdot\|_2$ to denote the L_2 norm, $\|x\| = x^\top x$. We also use $\|x\|_1 = \sum_i |x_i|$ and $\|x\|_\infty = \max_i |x_i|$. The **operator norm** of a matrix X with respect to norms $\|\cdot\|_P, \|\cdot\|_{P'}$ is defined as

$$\|X\|_{P\to P'} = \sup_{u:\|u\|_P=1} \|Xu\|_{P'} \, .$$

A function $f : \mathbb{R}^k \to \mathbb{R}^d$ is L-**Lipschitz** with respect to a norm $\|\cdot\|_P$, whose dual norm is $\|\cdot\|_D$, if for all $a, b \in \mathbb{R}^d$, we have

$$\|f(a) - f(b)\|_D \le L \,\|a - b\|_P.$$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is $(1/\gamma)$-**smooth** with respect to a norm $\|\cdot\|_P$ if it is differentiable and its gradient is $(1/\gamma)$-Lipschitz with respect to $\|\cdot\|_P$. An equivalent condition is that for all $a, b \in \mathbb{R}^d$, we have

$$f(a) \le f(b) + \nabla f(b)^\top (a - b) + \frac{1}{2\gamma}\|a - b\|_P^2.$$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is γ-**strongly convex** with respect to $\|\cdot\|_P$ if

$$f(a + b) \ge f(a) + \nabla f(a)^\top b + \frac{\gamma}{2}\|b\|_P^2.$$

It is well known that f is γ-strongly convex with respect to $\|\cdot\|_P$ if and only if f^* is $(1/\gamma)$-smooth with respect to the dual norm, $\|\cdot\|_D$.

The **dual problem** of (1) is

$$\max_{\alpha\in\mathbb{R}^{k\times n}} D(\alpha) \quad \text{where} \quad D(\alpha) = \left[\frac{1}{n}\sum_{i=1}^n -\phi_i^*(-\alpha_i) - \lambda g^*\left(\frac{1}{\lambda n}\sum_{i=1}^n X_i\alpha_i\right)\right], \tag{2}$$

where α_i is the i'th column of the matrix α, which forms a vector in $\mathbb{R}^k$.

We will assume that g is strongly convex which implies that $g^*(\cdot)$ is continuous differentiable. Using the Fenchel's duality theorem, and define

$$v(\alpha) = \frac{1}{\lambda n}\sum_{i=1}^n X_i\alpha_i \qquad \text{and} \qquad w(\alpha) = \nabla g^*(v(\alpha)), \tag{3}$$

we know that $w(\alpha^*) = w^*$, where α^* is an optimal solution of (2). Another implication of Fenchel's duality is that $P(w^*) = D(\alpha^*)$ which means that for all w and α, $P(w) \geq D(\alpha)$. The **duality gap**, defined as

$$P(w(\alpha)) - D(\alpha),$$

is an upper bound of both the **primal sub-optimality** $P(w(\alpha)) - P(w^*)$, and the **dual sub-optimality** $D(\alpha^*) - D(\alpha)$.

3 Main results

In this section we describe our algorithms and their analysis. We start in Sect. 3.1 with a description of our proximal stochastic dual coordinate ascent procedure (Prox-SDCA). Then, in Sect. 3.2 we show how to accelerate the method by calling Prox-SDCA on a sequence of problems with a strong regularization. Throughout the first two sections we assume that the loss functions are smooth. Finally, we discuss the case of Lipschitz loss functions in Sect. 3.3.

The proofs of the main acceleration theorem (Theorem 3) is given in Sect. 4. The rest of the proofs are provided in the "Appendix".

3.1 Proximal stochastic dual coordinate ascent

We now describe our proximal stochastic dual coordinate ascent procedure for solving (1). Our results in this subsection holds under the following assumption.

Assumption 1 *Assume that g is a 1-strongly convex function with respect to some norm $\|\cdot\|_{P'}$ and every ϕ_i being a $(1/\gamma)$-smooth function with respect to some other norm $\|\cdot\|_P$. The corresponding dual norms are denoted by $\|\cdot\|_{D'}$ and $\|\cdot\|_D$ respectively.*

The dual objective in (2) has a different dual vector associated with each example in the training set. At each iteration of dual coordinate ascent we only allow to change the i'th column of α, while the rest of the dual vectors are kept intact. We focus on a *randomized* version of dual coordinate ascent, in which at each round we choose which dual vector to update uniformly at random.

At step t, let $v^{(t-1)} = (\lambda n)^{-1} \sum_i X_i \alpha_i^{(t-1)}$ and let $w^{(t-1)} = \nabla g^*(v^{(t-1)})$. We will update the i-th dual variable $\alpha_i^{(t)} = \alpha_i^{(t-1)} + \Delta\alpha_i$, in a way that will lead to a sufficient increase of the dual objective. For the primal problem, this would lead to the update $v^{(t)} = v^{(t-1)} + (\lambda n)^{-1} X_i \Delta\alpha_i$, and therefore $w^{(t)} = \nabla g^*(v^{(t)})$ can also be written as

$$w^{(t)} = \operatorname*{argmax}_w \left[w^\top v^{(t)} - g(w) \right] = \operatorname*{argmin}_w \left[-w^\top \left(n^{-1} \sum_{i=1}^n X_i \alpha_i^{(t)} \right) + \lambda g(w) \right].$$

Note that this particular update is rather similar to the update step of proximal-gradient dual-averaging method (see for example Xiao [30]). The difference is on how $\alpha^{(t)}$ is updated.

The goal of dual ascent methods is to increase the dual objective as much as possible, and thus the optimal way to choose $\Delta\alpha_i$ would be to maximize the dual objective, namely, we shall let

$$\Delta\alpha_i = \operatorname*{argmax}_{\Delta\alpha_i \in \mathbb{R}^k} \left[-\frac{1}{n}\phi_i^*(-(\alpha_i + \Delta\alpha_i)) - \lambda g^*(v^{(t-1)} + (\lambda n)^{-1} X_i \Delta\alpha_i) \right].$$

However, for a complex $g^*(\cdot)$, this optimization problem may not be easy to solve. To simplify the optimization problem we can rely on the smoothness of g^* (with respect to a norm $\|\cdot\|_{D'}$) and instead of directly maximizing the dual objective function, we try to maximize the following proximal objective which is a lower bound of the dual objective:

$$\operatorname*{argmax}_{\Delta\alpha_i \in \mathbb{R}^k} \left[-\frac{1}{n}\phi_i^*(-(\alpha_i + \Delta\alpha_i)) - \lambda \left(\nabla g^*(v^{(t-1)})^\top (\lambda n)^{-1} X_i \Delta\alpha_i + \frac{1}{2}\|(\lambda n)^{-1} X_i \Delta\alpha_i\|_{D'}^2 \right) \right]$$
$$= \operatorname*{argmax}_{\Delta\alpha_i \in \mathbb{R}^k} \left[-\phi_i^*(-(\alpha_i + \Delta\alpha_i)) - w^{(t-1)\top} X_i \Delta\alpha_i - \frac{1}{2\lambda n}\|X_i \Delta\alpha_i\|_{D'}^2 \right].$$

In general, this optimization problem is still not necessarily simple to solve because ϕ^* may also be complex. We will thus also propose alternative update rules for $\Delta\alpha_i$ of the form $\Delta\alpha_i = s(-\nabla\phi_i(X_i^\top w^{(t-1)}) - \alpha_i^{(t-1)})$ for an appropriately chosen step size parameter $s > 0$. Our analysis shows that an appropriate choice of s still leads to a sufficient increase in the dual objective.

This leads to a number of options in Procedure Prox-SDCA of Fig. 1. The first option yields the largest dual increase, while other options yield smaller (but still sufficiently large) dual increase. For each loss function and regularizer, one can choose the option which yields the largest dual increase among the options which are easy to compute for the specific problem. For example, for the smoothed hinge-loss (with a Euclidean regularization), the first option has a closed form solution and therefore we will use it. In contrast, for logistic regression, the first two options has no closed form solution, but options III–V are easy to compute, so we can use Option III.

It should be pointed out that we can always pick $\Delta\alpha_i$ so that the dual objective is non-decreasing. In fact, if for a specific choice of $\Delta\alpha_i$, the dual objective decreases, we may simply set $\Delta\alpha_i = 0$. Therefore throughout the proof we will assume that the dual objective is non-decreasing whenever needed.

The theorems below provide upper bounds on the number of iterations required by our prox-SDCA procedure.

Theorem 1 *Consider Procedure Prox-SDCA as given in Fig. 1. Let α^* be an optimal dual solution and let $\epsilon > 0$. If Assumption 1 holds, then for every T such that*

$$T \geq \left(n + \frac{R^2}{\lambda\gamma}\right) \log\left(\left(n + \frac{R^2}{\lambda\gamma}\right) \cdot \frac{D(\alpha^*) - D(\alpha^{(0)})}{\epsilon}\right),$$

Procedure Proximal Stochastic Dual Coordinate Ascent:
Prox-SDCA($P, \epsilon, \alpha^{(0)}$)

Goal: Minimize $P(w) = \frac{1}{n}\sum_{i=1}^n \phi_i(X_i^\top w) + \lambda g(w)$
Input: Objective P, desired accuracy ϵ, initial dual solution $\alpha^{(0)}$ (default: $\alpha^{(0)} = 0$)
Assumptions:
$\forall i$, ϕ_i is $(1/\gamma)$-smooth w.r.t. $\|\cdot\|_P$ and let $\|\cdot\|_D$ be the dual norm of $\|\cdot\|_P$
g is 1-strongly convex w.r.t. $\|\cdot\|_{P'}$ and let $\|\cdot\|_{D'}$ be the dual norm of $\|\cdot\|_{P'}$
$\forall i$, $\|X_i\|_{D\to D'} \le R$
Initialize $v^{(0)} = \frac{1}{\lambda n}\sum_{i=1}^n X_i \alpha_i^{(0)}$, $w^{(0)} = \nabla g^*(0)$
Iterate: for $t = 1, 2, \ldots$
Randomly pick i
Find $\Delta\alpha_i$ using any of the following options
(or any other update that achieves a larger dual objective):
Option I:

$$\Delta\alpha_i = \operatorname*{argmax}_{\Delta\alpha_i}\left[-\phi_i^*(-(\alpha_i^{(t-1)} + \Delta\alpha_i)) - w^{(t-1)^\top} X_i \Delta\alpha_i - \frac{1}{2\lambda n}\|X_i \Delta\alpha_i\|_{D'}^2\right]$$

Option II:
Let $u = -\nabla\phi_i(X_i^\top w^{(t-1)})$ and $q = u - \alpha_i^{(t-1)}$

$$\text{Let } s = \operatorname*{argmax}_{s\in[0,1]}\left[-\phi_i^*(-(\alpha_i^{(t-1)} + sq)) - s\, w^{(t-1)^\top} X_i q - \frac{s^2}{2\lambda n}\|X_i q\|_{D'}^2\right]$$

Set $\Delta\alpha_i = sq$
Option III:
Same as Option II but replace the definition of s as follows:

$$\text{Let } s = \min\left(1, \frac{\phi_i(X_i^\top w^{(t-1)}) + \phi_i^*(-\alpha_i^{(t-1)}) + w^{(t-1)^\top} X_i \alpha_i^{(t-1)} + \frac{\gamma}{2}\|q\|_D^2}{\|q\|_D^2(\gamma + \frac{1}{\lambda n}\|X_i\|_{D\to D'}^2)}\right)$$

Option IV:
Same as Option III but replace $\|X_i\|_{D\to D'}^2$ in the definition of s with R^2
Option V:
Same as Option II but replace the definition of s to be $s = \frac{\lambda n\gamma}{R^2 + \lambda n\gamma}$
$\alpha_i^{(t)} \leftarrow \alpha_i^{(t-1)} + \Delta\alpha_i$ and for $j \ne i$, $\alpha_j^{(t)} \leftarrow \alpha_j^{(t-1)}$
$v^{(t)} \leftarrow v^{(t-1)} + (\lambda n)^{-1} X_i \Delta\alpha_i$
$w^{(t)} \leftarrow \nabla g^*(v^{(t)})$
Stopping condition:
Let $T_0 < t$ (default: $T_0 = t - n - \lceil \frac{R^2}{\lambda\gamma} \rceil$)
Averaging option:
Let $\bar\alpha = \frac{1}{t-T_0}\sum_{i=T_0+1}^t \alpha^{(i-1)}$ and $\bar w = \frac{1}{t-T_0}\sum_{i=T_0+1}^t w^{(i-1)}$
Random option:
Let $\bar\alpha = \alpha^{(i)}$ and $\bar w = w^{(i)}$ for some random $i \in T_0 + 1, \ldots, t$
Stop if $P(\bar w) - D(\bar\alpha) \le \epsilon$ and output $\bar w, \bar\alpha$, and $P(\bar w) - D(\bar\alpha)$

Fig. 1 The generic proximal stochastic dual coordinate ascent algorithm

we are guaranteed that $\mathbb{E}[P(w^{(T)}) - D(\alpha^{(T)})] \le \epsilon$. *Moreover, for every* T *such that*

$$T \ge \left(n + \left\lceil \frac{R^2}{\lambda\gamma} \right\rceil\right) \cdot \left(1 + \log\left(\frac{D(\alpha^*) - D(\alpha^{(0)})}{\epsilon}\right)\right),$$

let $T_0 = T - n - \lceil \frac{R^2}{\lambda\gamma} \rceil$, *then we are guaranteed that* $\mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] \leq \epsilon$.

We next give bounds that hold with high probability.

Theorem 2 *Consider Procedure Prox-SDCA as given in Fig. 1. Let* α^* *be an optimal dual solution, let* $\epsilon_D, \epsilon_P > 0$*, and let* $\delta \in (0, 1)$*. If Assumption 1 holds, then we have:*

1. *For every* T *such that*

$$T \geq \left\lceil \left(n + \frac{R^2}{\lambda\gamma}\right) \log\left(\frac{2(D(\alpha^*) - D(\alpha^{(0)}))}{\epsilon_D}\right)\right\rceil \cdot \left\lceil \log_2\left(\frac{1}{\delta}\right)\right\rceil,$$

we are guaranteed that with probability of at least $1 - \delta$ *it holds that* $D(\alpha^*) - D(\alpha^{(T)}) \leq \epsilon_D$.

2. *For every* T *such that*

$$T \geq \left\lceil \left(n + \frac{R^2}{\lambda\gamma}\right) \left(\log\left(n + \frac{R^2}{\lambda\gamma}\right) + \log\left(\frac{2(D(\alpha^*) - D(\alpha^{(0)}))}{\epsilon_P}\right)\right)\right\rceil \cdot \left\lceil \log_2\left(\frac{1}{\delta}\right)\right\rceil,$$

we are guaranteed that with probability of at least $1 - \delta$ *it holds that* $P(w^{(T)}) - D(\alpha^{(T)}) \leq \epsilon_P$.

3. *Let* T *be such that*

$$T \geq \left(n + \left\lceil \frac{R^2}{\lambda\gamma}\right\rceil\right) \cdot \left(1 + \left\lceil \log\left(\frac{2(D(\alpha^*) - D(\alpha^{(0)}))}{\epsilon_P}\right)\right\rceil\right) \cdot \left\lceil \log_2\left(\frac{2}{\delta}\right)\right\rceil,$$

and let $T_0 = T - n - \lceil \frac{R^2}{\lambda\gamma} \rceil$*. Suppose we choose* $\lceil \log_2(2/\delta) \rceil$ *values of* t *uniformly at random from* $T_0 + 1, \ldots, T$*, and then choose the single value of* t *from these* $\lceil \log_2(2/\delta) \rceil$ *values for which* $P(w^{(t)}) - D(\alpha^{(t)})$ *is minimal. Then, with probability of at least* $1 - \delta$ *we have that* $P(w^{(t)}) - D(\alpha^{(t)}) \leq \epsilon_P$.

The above theorem tells us that the runtime required to find an ϵ accurate solution, with probability of at least $1 - \delta$, is

$$O\left(d\left(n + \frac{R^2}{\lambda\gamma}\right) \cdot \log\left(\frac{D(\alpha^*) - D(\alpha^{(0)})}{\epsilon}\right) \cdot \log\left(\frac{1}{\delta}\right)\right). \tag{4}$$

This yields the following corollary.

Corollary 1 *The expected runtime required to minimize* P *up to accuracy* ϵ *is*

$$O\left(d\left(n + \frac{R^2}{\lambda\gamma}\right) \cdot \log\left(\frac{D(\alpha^*) - D(\alpha^{(0)})}{\epsilon}\right)\right).$$

Proof We have shown that with a runtime of $O\left(d\left(n+\frac{R^2}{\lambda\gamma}\right)\cdot\log\left(\frac{2(D(\alpha^*)-D(\alpha^{(0)}))}{\epsilon}\right)\right)$ we can find an ϵ accurate solution with probability of at least 1/2. Therefore, we can run the procedure for this amount of time and check if the duality gap is smaller than ϵ. If yes, we are done. Otherwise, we would restart the process. Since the probability of success is 1/2 we have that the average number of restarts we need is 2, which concludes the proof. □

3.2 Acceleration

The Prox-SDCA procedure described in the previous subsection has the iteration bound of $\tilde{O}\left(n+\frac{R^2}{\lambda\gamma}\right)$. This is a nearly linear runtime whenever the condition number, $R^2/(\lambda\gamma)$, is $O(n)$. In this section we show how to improve the dependence on the condition number by an acceleration procedure. In particular, throughout this section we assume that $10\,n<\frac{R^2}{\lambda\gamma}$. We note that the acceleration procedure of this section does not improve performance of Prox-SDCA when the condition number is small (that is, when $R^2/(\lambda\gamma))=O(n)$), and in this scenario we are unaware of any method that can provide improvement. We further assume throughout this subsection that the regularizer, g, is 1-strongly convex with respect to the Euclidean norm, i.e. $\|u\|_{P'}=\|\cdot\|_2$. This also implies that $\|u\|_{D'}$ is the Euclidean norm. A generalization of the acceleration technique for strongly convex regularizers with respect to general norms is left to future work.

The main idea of the acceleration procedure is to iteratively run the Prox-SDCA procedure, where at iteration t we call Prox-SDCA with the modified objective, $\tilde{P}_t(w)=P(w)+\frac{\kappa}{2}\|w-y^{(t-1)}\|^2$, where κ is a relatively large regularization parameter and the regularization is centered around the vector

$$y^{(t-1)}=w^{(t-1)}+\beta(w^{(t-1)}-w^{(t-2)})$$

for some $\beta\in(0,1)$. That is, our regularization is centered around the previous solution plus a "momentum term" $\beta(w^{(t-1)}-w^{(t-2)})$.

A pseudo-code of the algorithm is given in Fig. 2. Note that all the parameters of the algorithm are determined by our theory.

Remark 1 In the pseudo-code below, we specify the parameters based on our theoretical derivation. In our experiments, we found out that this choice of parameters also work very well in practice. However, we also found out that the algorithm is not very sensitive to the choice of parameters. For example, we found out that running $5n$ iterations of Prox-SDCA (that is, 5 epochs over the data), without checking the stopping condition, also works very well.

The main theorem is the following.

Theorem 3 *Consider the accelerated Prox-SDCA algorithm given in Fig. 2.*

- *Correctness: when the algorithm terminates we have that* $P(w^{(t)})-P(w^*)\leq\epsilon$.
- *Runtime:*

Procedure Accelerated Prox-SDCA

Goal: Minimize $P(w) = \frac{1}{n}\sum_{i=1}^{n} \phi_i(X_i^\top w) + \lambda g(w)$
Input: Target accuracy ϵ (only used in the stopping condition)
Assumptions:
$\forall i$, ϕ_i is $(1/\gamma)$-smooth w.r.t. $\|\cdot\|_P$ and let $\|\cdot\|_D$ be the dual norm of $\|\cdot\|_P$
g is 1-strongly convex w.r.t. $\|\cdot\|_2$
$\forall i$, $\|X_i\|_{D\to 2} \le R$
$\frac{R^2}{\gamma\lambda} > 10\, n$ (otherwise, solve the problem using vanilla Prox-SDCA)
Define $\kappa = \frac{R^2}{\gamma n} - \lambda$, $\mu = \lambda/2$, $\rho = \mu + \kappa$, $\eta = \sqrt{\mu/\rho}$, $\beta = \frac{1-\eta}{1+\eta}$,
Initialize $y^{(1)} = w^{(1)} = 0$, $\alpha^{(1)} = 0$, $\xi_1 = (1+\eta^{-2})(P(0) - D(0))$
Iterate: for $t = 2, 3, \ldots$
Let $\tilde{P}_t(w) = \frac{1}{n}\sum_{i=1}^{n} \phi_i(X_i^\top w) + \tilde{\lambda}\tilde{g}_t(w)$
where $\tilde{\lambda}\tilde{g}_t(w) = \lambda g(w) + \frac{\kappa}{2}\|w\|_2^2 - \kappa\, w^\top y^{(t-1)}$
Call $(w^{(t)}, \alpha^{(t)}, \epsilon_t) =$ Prox-SDCA $\left(\tilde{P}_t, \frac{\eta}{2(1+\eta^{-2})}\xi_{t-1}, \alpha^{(t-1)}\right)$
Let $y^{(t)} = w^{(t)} + \beta(w^{(t)} - w^{(t-1)})$
Let $\xi_t = (1-\eta/2)^{t-1}\, \xi_1$
Stopping conditions: break and return $w^{(t)}$ if one of the following conditions hold:
1. $t \ge 1 + \frac{2}{\eta}\log(\xi_1/\epsilon)$
2. $(1+\rho/\mu)\epsilon_t + \frac{\rho\kappa}{2\mu}\|w^{(t)} - y^{(t-1)}\|^2 \le \epsilon$

Fig. 2 The accelerated Prox-SDCA algorithm

– *The number of outer iterations is at most*

$$1 + \frac{2}{\eta}\log(\xi_1/\epsilon) \;\le\; 1 + \sqrt{\frac{8R^2}{\lambda\gamma n}}\left(\log\left(\frac{2R^2}{\lambda\gamma n}\right) + \log\left(\frac{P(0) - D(0)}{\epsilon}\right)\right).$$

– *Each outer iteration involves a single call to Prox-SDCA, and the averaged runtime required by each such call is*

$$O\left(d\, n \log\left(\frac{R^2}{\lambda\gamma n}\right)\right).$$

By a straightforward amplification argument we obtain that for every $\delta \in (0, 1)$ the total runtime required by accelerated Prox-SDCA to guarantee an ϵ-accurate solution with probability of at least $1 - \delta$ is

$$O\left(d\sqrt{\frac{nR^2}{\lambda\,\gamma}}\,\log\left(\frac{R^2}{\lambda\,\gamma\,n}\right)\left(\log\left(\frac{R^2}{\lambda\gamma n}\right) + \log\left(\frac{P(0) - D(0)}{\epsilon}\right)\right)\log\left(\frac{1}{\delta}\right)\right).$$

3.3 Non-smooth, Lipschitz, loss functions

So far we have assumed that for every i, ϕ_i is a $(1/\gamma)$-smooth function. We now consider the case in which ϕ_i might be non-smooth, and even non-differentiable, but it is L-Lipschitz.

Following Nesterov [18], we apply a "smoothing" technique. We first observe that if ϕ is L-Lipschitz function then the domain of ϕ^* is in the ball of radius L. The corresponding lemma is stated below. A simpler version (with real valued function ϕ) can also be found in [26].

Lemma 1 *Let $\phi : \mathbb{R}^k \to \mathbb{R}$ be an L-Lipschitz function w.r.t. a norm $\|\cdot\|_P$ and let $\|\cdot\|_D$ be the dual norm. Then, for any $\alpha \in \mathbb{R}^k$ s.t. $\|\alpha\|_D > L$ we have that $\phi^*(\alpha) = \infty$.*

The proof is similar to the proof of the corresponding lemma in [26], and the interested reader can find it in Shalev-Shwartz and Zhang [25].

This observation allows us to smooth L-Lipschitz functions by adding regularization to their conjugate. In particular, the following lemma generalizes Lemma 2.5 in [28].

Lemma 2 *Let ϕ be a proper, convex, L-Lipschitz function w.r.t. a norm $\|\cdot\|_P$, let $\|\cdot\|_D$ be the dual norm, and let ϕ^* be the conjugate of ϕ. Assume that $\|\cdot\|_2 \le \|\cdot\|_D$. Define $\tilde{\phi}^*(\alpha) = \phi^*(\alpha) + \frac{\gamma}{2}\|\alpha\|_2^2$ and let $\tilde{\phi}$ be the conjugate of $\tilde{\phi}^*$. Then, $\tilde{\phi}$ is $(1/\gamma)$-smooth w.r.t. the Euclidean norm and*

$$\forall a, 0 \le \phi(a) - \tilde{\phi}(a) \le \gamma L^2/2.$$

Proof The fact that $\tilde{\phi}$ is $(1/\gamma)$-smooth follows directly from the fact that $\tilde{\phi}^*$ is γ-strongly convex. For the second claim note that

$$\tilde{\phi}(a) = \sup_b \left[b^\top a - \phi^*(b) - \frac{\gamma}{2}\|b\|_2^2 \right] \le \sup_b \left[b^\top a - \phi^*(b) \right] = \phi(a)$$

and

$$\begin{aligned}
\tilde{\phi}(a) &= \sup_b \left[b^\top a - \phi^*(b) - \frac{\gamma}{2}\|b\|_2^2 \right] = \sup_{b:\|b\|_D \le L} \left[b^\top a - \phi^*(b) - \frac{\gamma}{2}\|b\|_2^2 \right] \\
&\ge \sup_{b:\|b\|_D \le L} \left[b^\top a - \phi^*(b) - \frac{\gamma}{2}\|b\|_D^2 \right] \ge \sup_{b:\|b\|_D \le L} \left[b^\top a - \phi^*(b) \right] - \frac{\gamma}{2}L^2 \\
&= \phi(a) - \frac{\gamma}{2}L^2.
\end{aligned}$$

□

Remark 2 It is also possible to smooth using different regularization functions which are strongly convex with respect to other norms. See Nesterov [18] for discussion.

4 Proof of Theorem 3

The first claim of the theorem is that when the procedure stops we have $P(w^{(t)}) - P(w^*) \le \epsilon$. We therefore need to show that each stopping condition guarantees that $P(w^{(t)}) - P(w^*) \le \epsilon$.

For the second stopping condition, recall that $w^{(t)}$ is an ϵ_t-accurate minimizer of $P(w) + \frac{\kappa}{2}\|w - y^{(t-1)}\|^2$, and hence by Lemma 3 below (with $z = w^*$, $w^+ = w^{(t)}$, and $y = y^{(t-1)}$):

$$\begin{aligned} P(w^*) &\ge P(w^{(t)}) + Q_\epsilon(w^*; w^{(t)}, y^{(t-1)}) \\ &\ge P(w^{(t)}) - \frac{\rho\kappa}{2\mu}\|y^{(t-1)} - w^{(t)}\|^2 - (1 + \rho/\mu)\epsilon_t. \end{aligned}$$

It is left to show that the first stopping condition is correct, namely, to show that after $1 + \frac{2}{\eta}\log(\xi_1/\epsilon)$ iterations the algorithm must converge to an ϵ-accurate solution. Observe that the definition of ξ_t yields that $\xi_t = (1 - \eta/2)^{t-1}\xi_1 \le e^{-\eta(t-1)/2}\xi_1$. Therefore, to prove that the first stopping condition is valid, it suffices to show that for every t, $P(w^{(t)}) - P(w^*) \le \xi_t$.

Recall that at each outer iteration of the accelerated procedure, we approximately minimize an objective of the form

$$P(w; y) = P(w) + \frac{\kappa}{2}\|w - y\|^2.$$

Of course, minimizing $P(w; y)$ is not the same as minimizing $P(w)$. Our first lemma shows that for every y, if w^+ is an ϵ-accurate minimizer of $P(w; y)$ then we can derive a lower bound on $P(w)$ based on $P(w^+)$ and a convex quadratic function of w.

The essence of the proof relies on an adaption of Nesterov's estimate sequence technique, and the necessary properties of the estimate sequence $h_t(z)$ are given in Lemma 4. Its proof requires a forward-looking inequality stated below.

Lemma 3 *Let $\mu = \lambda/2$ and $\rho = \mu + \kappa$. Let w^+ be a vector such that $P(w^+; y) \le \min_w P(w, y) + \epsilon$. Then, for every z,*

$$P(z) \ge P(w^+) + Q_\epsilon(z; w^+, y),$$

where

$$Q_\epsilon(z; w^+, y) = \frac{\mu}{2}\left\| z - \left(y - \tfrac{\rho}{\mu}(y - w^+)\right)\right\|^2 - \frac{\rho\kappa}{2\mu}\|y - w^+\|^2 - (1 + \rho/\mu)\epsilon.$$

Proof Denote

$$\Psi(w) = P(w) - \frac{\mu}{2}\|w\|^2.$$

We can write

$$\frac{1}{2}\|w\|^2 = \frac{1}{2}\|y\|^2 + y^\top(w-y) + \frac{1}{2}\|w-y\|^2.$$

It follows that

$$P(w) = \Psi(w) + \frac{\mu}{2}\|w\|^2 = \Psi(w) + \frac{\mu}{2}\|y\|^2 + \mu\, y^\top(w-y) + \frac{\mu}{2}\|w-y\|^2.$$

Therefore, we can rewrite $P(w; y)$ as:

$$P(w; y) = \Psi(w) + \frac{\mu}{2}\|y\|^2 + \mu\, y^\top(w-y) + \frac{\rho}{2}\|w-y\|^2.$$

Let $\tilde{w} = \operatorname{argmin}_w P(w; y)$. Therefore, the gradient[2] of $P(w; y)$ w.r.t. w vanishes at $\tilde{w}$, which yields

$$\nabla\Psi(\tilde{w}) + \mu y + \rho(\tilde{w} - y) = 0 \Rightarrow \nabla\Psi(\tilde{w}) + \mu y = \rho(y - \tilde{w}).$$

By the μ-strong convexity of Ψ we have that for every z,

$$\Psi(z) \geq \Psi(\tilde{w}) + \nabla\Psi(\tilde{w})^\top(z - \tilde{w}) + \frac{\mu}{2}\|z - \tilde{w}\|^2.$$

Therefore,

$$\begin{aligned}
P(z) &= \Psi(z) + \frac{\mu}{2}\|y\|^2 + \mu\, y^\top(z-y) + \frac{\mu}{2}\|z-y\|^2 \\
&\geq \Psi(\tilde{w}) + \nabla\Psi(\tilde{w})^\top(z-\tilde{w}) \\
&\quad + \frac{\mu}{2}\|z-\tilde{w}\|^2 + \frac{\mu}{2}\|y\|^2 + \mu\, y^\top(z-y) + \frac{\mu}{2}\|z-y\|^2 \\
&= P(\tilde{w}; y) - \frac{\rho}{2}\|\tilde{w}-y\|^2 + \nabla\Psi(\tilde{w})^\top(z-\tilde{w}) + \mu\, y^\top(z-\tilde{w}) \\
&\quad + \frac{\mu}{2}\left(\|z-\tilde{w}\|^2 + \|z-y\|^2\right) \\
&= P(\tilde{w}; y) - \frac{\rho}{2}\|\tilde{w}-y\|^2 + \rho(y-\tilde{w})^\top(z-\tilde{w}) + \frac{\mu}{2}\left(\|z-\tilde{w}\|^2 + \|z-y\|^2\right) \\
&= P(\tilde{w}; y) + \frac{\rho}{2}\|\tilde{w}-y\|^2 + \rho(y-\tilde{w})^\top(z-y) + \frac{\mu}{2}\left(\|z-\tilde{w}\|^2 + \|z-y\|^2\right).
\end{aligned}$$

[2] If the regularizer $g(w)$ in the definition of $P(w)$ is non-differentiable, we can replace $\nabla\Psi(\tilde{w})$ with an appropriate sub-gradient of Ψ at $\tilde{w}$. It is easy to verify that the proof is still valid.

In addition, by standard algebraic manipulations,

$$
\begin{aligned}
&\frac{\rho}{2}\|\tilde{w}-y\|^2+\rho(y-\tilde{w})^\top(z-y)+\frac{\mu}{2}\|z-\tilde{w}\|^2\\
&\quad-\left(\frac{\rho}{2}\|w^+-y\|^2+\rho(y-w^+)^\top(z-y)+\frac{\mu}{2}\|z-w^+\|^2\right)\\
&=\left(\rho(w^+-y)-\rho(z-y)+\mu(w^+-z)\right)^\top(\tilde{w}-w^+)+\frac{\rho+\mu}{2}\|\tilde{w}-w^+\|^2\\
&=(\rho+\mu)(w^+-z)^\top(\tilde{w}-w^+)+\frac{\rho+\mu}{2}\|\tilde{w}-w^+\|^2\\
&=\frac{1}{2}\left\|\sqrt{\mu}(w^+-z)+\frac{\rho+\mu}{\sqrt{\mu}}(\tilde{w}-w^+)\right\|^2\\
&\quad-\frac{\mu}{2}\|z-w^+\|^2-\frac{(\rho+\mu)^2}{2\mu}\|\tilde{w}-w^+\|^2+\frac{\rho+\mu}{2}\|\tilde{w}-w^+\|^2\\
&\geq-\frac{\mu}{2}\|z-w^+\|^2-\frac{\rho(\rho+\mu)}{2\mu}\|\tilde{w}-w^+\|^2.
\end{aligned}
$$

Since $P(\cdot;y)$ is $(\rho+\mu)$-strongly convex and $\tilde{w}$ minimizes $P(\cdot;y)$, we have that for every w^+ it holds that $\frac{\rho+\mu}{2}\|\tilde{w}-w^+\|^2\leq P(w^+;y)-P(\tilde{w};y)$. Combining all the above and using the fact that for every w,y, $P(w;y)\geq P(w)$, we obtain that for every w^+,

$$
\begin{aligned}
P(z)\geq P(w^+)&+\frac{\rho}{2}\|w^+-y\|^2+\rho(y-w^+)^\top(z-y)+\frac{\mu}{2}\|z-y\|^2\\
&-\left(1+\frac{\rho}{\mu}\right)\left(P(w^+;y)-P(\tilde{w};y)\right).
\end{aligned}
$$

Finally, using the assumption $P(w^+;y)\leq\min_w P(w;y)+\epsilon$ we conclude our proof. □

We saw that the quadratic function $P(w^+)+Q_\epsilon(z;w^+,y)$ lower bounds the function P everywhere. Therefore, any convex combination of such functions would form a quadratic function which lower bounds P. In particular, the algorithm (implicitly) maintains a sequence of quadratic functions, $h_1,h_2,\ldots$, defined as follows. Choose $\eta\in(0,1)$ and a sequence $y^{(1)},y^{(2)},\ldots$ that will be specified later. Define,

$$
h_1(z)=P(0)+Q_{P(0)-D(0)}(z;0,0)=P(0)+\frac{\mu}{2}\|z\|^2-(1+\rho/\mu)(P(0)-D(0)),
$$

and for $t\geq 1$,

$$
h_{t+1}(z)=(1-\eta)h_t(z)+\eta(P(w^{(t+1)})+Q_{\epsilon_{t+1}}(z;w^{(t+1)},y^{(t)})).
$$

The following simple lemma shows that for every $t\geq 1$ and z, $h_t(z)$ lower bounds $P(z)$.

Lemma 4 *Let $\eta \in (0, 1)$ and let $y^{(1)}, y^{(2)}, \ldots$ be any sequence of vectors. Assume that $w^{(1)} = 0$ and for every $t \geq 1$, $w^{(t+1)}$ satisfies $P(w^{(t+1)}; y^{(t)}) \leq \min_w P(w; y^{(t)}) + \epsilon_{t+1}$. Then, for every $t \geq 1$ and every vector z we have*

$$h_t(z) \leq P(z).$$

Proof The proof is by induction. For $t = 1$, observe that $P(0; 0) = P(0)$ and that for every w we have $P(w; 0) \geq P(w) \geq D(0)$. This yields $P(0; 0) - \min_w P(w; 0) \leq P(0) - D(0)$. The claim now follows directly from Lemma 3. Next, for the inductive step, assume the claim holds for some $t - 1 \geq 1$ and let us prove it for t. By the recursive definition of h_t and by using Lemma 3 we have

$$\begin{aligned} h_t(z) &= (1-\eta)h_{t-1}(z) + \eta(P(w^{(t)}) + Q_{\epsilon_t}(z; w^{(t)}, y^{(t-1)})) \\ &\leq (1-\eta)h_{t-1}(z) + \eta P(z). \end{aligned}$$

Using the inductive assumption we obtain that the right-hand side of the above is upper bounded by $(1-\eta)P(z) + \eta P(z) = P(z)$, which concludes our proof. □

The more difficult part of the proof is to show that for every $t \geq 1$,

$$P(w^{(t)}) \leq \min_w h_t(w) + \xi_t.$$

If this holds true, then we would immediately get that for every w^*,

$$P(w^{(t)}) - P(w^*) \leq P(w^{(t)}) - h_t(w^*) \leq P(w^{(t)}) - \min_w h_t(w) \leq \xi_t.$$

This will conclude the proof of the first part of Theorem 3, since $\xi_t = \xi_1(1-\eta/2)^{t-1} \leq \xi_1 \, e^{-(t-1)\eta/2}$, and therefore, $1 + \frac{2}{\eta}\log(\xi_1/\epsilon)$ iterations suffice to guarantee that $P(w^{(t)}) - P(w^*) \leq \epsilon$.

Define

$$v^{(t)} = \operatorname*{argmin}_w h_t(w).$$

Let us construct an explicit formula for $v^{(t)}$. Clearly, $v^{(1)} = 0$. Assume that we have calculated $v^{(t)}$ and let us calculate $v^{(t+1)}$. Note that h_t is a quadratic function which is minimized at $v^{(t)}$. Furthermore, it is easy to see that for every t, h_t is μ-strongly convex quadratic function. Therefore,

$$h_t(z) = h_t(v^{(t)}) + \frac{\mu}{2}\|z - v^{(t)}\|^2 .$$

By the definition of h_{t+1} we obtain that

$$\begin{aligned} h_{t+1}(z) = (1-\eta)\left(h_t(v^{(t)}) + \frac{\mu}{2}\|z - v^{(t)}\|^2\right) + \eta(P(w^{(t+1)}) \\ + Q_{\epsilon_{t+1}}\left(z; w^{(t+1)}, y^{(t)}\right)\right). \end{aligned}$$

Since the gradient of $h_{t+1}(z)$ at $v^{(t+1)}$ should be zero, we obtain that $v^{(t+1)}$ should satisfy

$$(1-\eta)\mu(v^{(t+1)} - v^{(t)}) + \eta\mu\left(v^{(t+1)} - (y^{(t)} - \tfrac{\rho}{\mu}(y^{(t)} - w^{(t+1)}))\right) = 0$$

Rearranging, we obtain

$$v^{(t+1)} = (1-\eta)v^{(t)} + \eta(y^{(t)} - \tfrac{\rho}{\mu}(y^{(t)} - w^{(t+1)})). \tag{5}$$

Getting back to our second phase of the proof, we need to show that for every t we have $P(w^{(t)}) \le h_t(v^{(t)}) + \xi_t$. We do so by induction. For the case $t = 1$ we have

$$P(w^{(1)}) - h_1(v^{(1)}) = P(0) - h_1(0) = (1 + \rho/\mu)(P(0) - D(0)) = \xi_1.$$

For the induction step, assume the claim holds for $t \ge 1$ and let us prove it for $t + 1$. We use the shorthands,

$$Q_t(z) = Q_{\epsilon_t}(z; w^{(t)}, y^{(t-1)}) \quad \text{and} \quad \psi_t(z) = Q_t(z) + P(w^{(t)}).$$

Let us rewrite $h_{t+1}(v^{(t+1)})$ as

$$\begin{aligned} h_{t+1}(v^{(t+1)}) &= (1-\eta)h_t(v^{(t+1)}) + \eta\psi_{t+1}(v^{(t+1)}) \\ &= (1-\eta)\left(h_t(v^{(t)}) + \frac{\mu}{2}\|v^{(t)} - v^{(t+1)}\|^2\right) + \eta\psi_{t+1}(v^{(t+1)}). \end{aligned}$$

By the inductive assumption we have $h_t(v^{(t)}) \ge P(w^{(t)}) - \xi_t$ and by Lemma 3 we have $P(w^{(t)}) \ge \psi_{t+1}(w^{(t)})$. Therefore,

$$\begin{aligned} h_{t+1}(v^{(t+1)}) &\ge (1-\eta)\left(\psi_{t+1}(w^{(t)}) - \xi_t + \frac{\mu}{2}\|v^{(t)} - v^{(t+1)}\|^2\right) + \eta\psi_{t+1}(v^{(t+1)}) \\ &= \frac{(1-\eta)\mu}{2}\|v^{(t)} - v^{(t+1)}\|^2 + \eta\psi_{t+1}(v^{(t+1)}) \\ &\quad + (1-\eta)\psi_{t+1}(w^{(t)}) - (1-\eta)\xi_t. \end{aligned} \tag{6}$$

Next, note that we can rewrite

$$\begin{aligned} Q_{t+1}(z) = \frac{\mu}{2}\|z - y^{(t)}\|^2 + \rho(z - y^{(t)})^\top (y^{(t)} - w^{(t+1)}) \\ + \frac{\rho}{2}\|y^{(t)} - w^{(t+1)}\|^2 - (1 + \rho/\mu)\epsilon_{t+1}. \end{aligned}$$

Therefore,

$$\begin{aligned}
&\eta\psi_{t+1}(v^{(t+1)}) + (1-\eta)\psi_{t+1}(w^{(t)}) - P(w^{(t+1)}) + (1+\rho/\mu)\epsilon_{t+1} \\
&\quad = \frac{\eta\mu}{2}\|v^{(t+1)} - y^{(t)}\|^2 + \frac{(1-\eta)\mu}{2}\|w^{(t)} - y^{(t)}\|^2 \\
&\qquad + \rho(\eta v^{(t+1)} + (1-\eta)w^{(t)} - y^{(t)})^\top (y^{(t)} - w^{(t+1)}) \\
&\qquad + \frac{\rho}{2}\|y^{(t)} - w^{(t+1)}\|^2
\end{aligned} \tag{7}$$

So far we did not specify η and $y^{(t)}$ (except $y^{(0)} = 0$). We next set

$$\eta = \sqrt{\mu/\rho} \quad \text{and} \quad \forall t \geq 1,\ \ y^{(t)} = (1+\eta)^{-1}(\eta v^{(t)} + w^{(t)}).$$

This choices guarantees that (see (5))

$$\begin{aligned}
\eta v^{(t+1)} + (1-\eta)w^{(t)} &= \eta(1-\eta)v^{(t)} + \eta^2\left(1-\frac{\rho}{\mu}\right)y^{(t)} + \eta^2\frac{\rho}{\mu}w^{(t+1)} + (1-\eta)w^{(t)} \\
&= w^{(t+1)} + (1-\eta)\left[\eta v^{(t)} + \frac{\eta^2\left(1-\frac{\rho}{\mu}\right)}{1-\eta}y^{(t)} + w^{(t)}\right] \\
&= w^{(t+1)} + (1-\eta)\left[\eta v^{(t)} - \frac{1-\eta^2}{1-\eta}y^{(t)} + w^{(t)}\right] \\
&= w^{(t+1)} + (1-\eta)\left[\eta v^{(t)} - (1+\eta)y^{(t)} + w^{(t)}\right] \\
&= w^{(t+1)}.
\end{aligned}$$

We also observe that $\epsilon_{t+1} \leq \frac{\eta\xi_t}{2(1+\eta^{-2})}$ which implies that $(1+\rho/\mu)\epsilon_{t+1} + (1-\eta)\xi_t \leq (1-\eta/2)\xi_t = \xi_{t+1}$. Combining the above with (6) and (7), and rearranging terms, we obtain that

$$\begin{aligned}
&h_{t+1}(v^{(t+1)}) - P(w^{(t+1)}) + \xi_{t+1} - \frac{(1-\eta)\mu}{2}\|w^{(t)} - y^{(t)}\|^2 \\
&\quad \geq \frac{(1-\eta)\mu}{2}\|v^{(t)} - v^{(t+1)}\|^2 + \frac{\eta\mu}{2}\|v^{(t+1)} - y^{(t)}\|^2 - \frac{\rho}{2}\|y^{(t)} - w^{(t+1)}\|^2\,.
\end{aligned}$$

Next, observe that $\rho\eta^2 = \mu$ and that by (5) we have

$$y^{(t)} - w^{(t+1)} = \eta\left[\eta y^{(t)} + (1-\eta)v^{(t)} - v^{(t+1)}\right].$$

We therefore obtain that

$$\begin{aligned} & h_{t+1}(v^{(t+1)}) - P(w^{(t+1)}) + \xi_{t+1} - \frac{(1-\eta)\mu}{2}\|w^{(t)} - y^{(t)}\|^2 \\ & \geq \frac{(1-\eta)\mu}{2}\|v^{(t)} - v^{(t+1)}\|^2 + \frac{\eta\mu}{2}\|y^{(t)} - v^{(t+1)}\|^2 \\ & \quad - \frac{\mu}{2}\|\eta y^{(t)} + (1-\eta)v^{(t)} - v^{(t+1)}\|^2 . \end{aligned}$$

The right-hand side of the above is non-negative because of the convexity of the function $f(z) = \frac{\mu}{2}\|z - v^{(t+1)}\|^2$, which yields

$$P(w^{(t+1)}) \leq h_{t+1}(v^{(t+1)}) + \xi_{t+1} - \frac{(1-\eta)\mu}{2}\|w^{(t)} - y^{(t)}\|^2 \leq h_{t+1}(v^{(t+1)}) + \xi_{t+1}.$$

This concludes our inductive argument.

Proving the "runtime" part of Theorem 3 We next show that each call to Prox-SDCA will terminate quickly. By the definition of κ we have that

$$\frac{R^2}{(\kappa + \lambda)\gamma} = n.$$

Therefore, based on Corollary 1 we know that the averaged runtime at iteration t is

$$O\left(d\, n \log\left(\frac{\tilde{D}_t(\alpha^*) - \tilde{D}_t(\alpha^{(t-1)})}{\frac{\eta}{2(1+\eta^{-2})}\xi_{t-1}}\right)\right).$$

The following lemma bounds the initial dual sub-optimality at iteration $t \geq 4$. Similar arguments will yield a similar result for $t < 4$.

Lemma 5

$$\tilde{D}_t(\alpha^*) - \tilde{D}_t(\alpha^{(t-1)}) \leq \epsilon_{t-1} + \frac{36\kappa}{\lambda}\xi_{t-3} .$$

Proof Define $\tilde{\lambda} = \lambda + \kappa$, $f(w) = \frac{\lambda}{\tilde{\lambda}}g(w) + \frac{\kappa}{2\tilde{\lambda}}\|w\|^2$, and $\tilde{g}_t(w) = f(w) - \frac{\kappa}{\tilde{\lambda}}w^\top y^{(t-1)}$. Note that $\tilde{\lambda}$ does not depend on t and therefore $v(\alpha) = \frac{1}{n\tilde{\lambda}}\sum_i X_i\alpha_i$ is the same for every t. Let,

$$\tilde{P}_t(w) = \frac{1}{n}\sum_{i=1}^{n}\phi_i(X_i^\top w) + \tilde{\lambda}\tilde{g}_t(w).$$

We have

$$\tilde{P}_t(w^{(t-1)}) = \tilde{P}_{t-1}(w^{(t-1)}) + \kappa w^{(t-1)\top}(y^{(t-2)} - y^{(t-1)}). \tag{8}$$

Since

$$\tilde{g}_t^*(\theta) = \max_w w^\top \left(\theta + \frac{\kappa}{\tilde{\lambda}} y^{(t-1)}\right) - f(w) = f^* \left(\theta + \frac{\kappa}{\tilde{\lambda}} y^{(t-1)}\right),$$

we obtain that the dual problem is

$$\tilde{D}_t(\alpha) = -\frac{1}{n} \sum_i \phi_i^*(-\alpha_i) - \tilde{\lambda} f^* \left(v(\alpha) + \frac{\kappa}{\tilde{\lambda}} y^{(t-1)}\right)$$

Let $z = \frac{\kappa}{\tilde{\lambda}}(y^{(t-1)} - y^{(t-2)})$, then, by the smoothness of f^* we have

$$\begin{aligned} f^* \left(v(\alpha) + \frac{\kappa}{\tilde{\lambda}} y^{(t-1)}\right) = f^* \left(v(\alpha) + \frac{\kappa}{\tilde{\lambda}} y^{(t-2)} + z\right) &\le f^* \left(v(\alpha) + \frac{\kappa}{\tilde{\lambda}} y^{(t-2)}\right) \\ &+ \nabla f^* \left(v(\alpha) + \frac{\kappa}{\tilde{\lambda}} y^{(t-2)}\right)^\top z + \frac{1}{2}\|z\|^2. \end{aligned}$$

Applying this for $\alpha^{(t-1)}$ and using $w^{(t-1)} = \nabla \tilde{g}_{t-1}^*(v(\alpha^{(t-1)})) = \nabla f^*(v(\alpha^{(t-1)}) + \frac{\kappa}{\tilde{\lambda}} y^{(t-2)})$, we obtain

$$f^*(v(\alpha^{(t-1)}) + \frac{\kappa}{\tilde{\lambda}} y^{(t-1)}) \le f^*(v(\alpha^{(t-1)}) + \frac{\kappa}{\tilde{\lambda}} y^{(t-2)}) + w^{(t-1)\top} z + \frac{1}{2}\|z\|^2.$$

It follows that

$$-\tilde{D}_t(\alpha^{(t-1)}) + \tilde{D}_{t-1}(\alpha^{(t-1)}) \le \kappa w^{(t-1)\top}(y^{(t-1)} - y^{(t-2)}) + \frac{\kappa^2}{2\tilde{\lambda}} \|y^{(t-1)} - y^{(t-2)}\|^2.$$

Combining the above with (8), we obtain that

$$\tilde{P}_t(w^{(t-1)}) - \tilde{D}_t(\alpha^{(t-1)}) \le \tilde{P}_{t-1}(w^{(t-1)}) - \tilde{D}_{t-1}(\alpha^{(t-1)}) + \frac{\kappa^2}{2\tilde{\lambda}} \|y^{(t-1)} - y^{(t-2)}\|^2.$$

Since $\tilde{P}_t(w^{(t-1)}) \ge \tilde{D}_t(\alpha^*)$ and since $\tilde{\lambda} \ge \kappa$ we get that

$$\tilde{D}_t(\alpha^*) - \tilde{D}_t(\alpha^{(t-1)}) \le \epsilon_{t-1} + \frac{\kappa}{2} \|y^{(t-1)} - y^{(t-2)}\|^2.$$

Next, we bound $\|y^{(t-1)} - y^{(t-2)}\|^2$. We have

$$\begin{aligned} \|y^{(t-1)} - y^{(t-2)}\| &= \|w^{(t-1)} - w^{(t-2)} + \beta(w^{(t-1)} - w^{(t-2)} - w^{(t-2)} + w^{(t-3)})\| \\ &\le 3 \max_{i \in \{1,2\}} \|w^{(t-i)} - w^{(t-i-1)}\|, \end{aligned}$$

where we used the triangle inequality and $\beta < 1$. By strong convexity of P we have, for every i,

$$\|w^{(i)} - w^*\| \le \sqrt{\frac{P(w^{(i)}) - P(w^*)}{\lambda/2}} \le \sqrt{\frac{\xi_i}{\lambda/2}},$$

which implies

$$\|w^{(t-i)} - w^{(t-i-1)}\| \le \|w^{(t-i)} - w^*\| + \|w^* - w^{(t-i-1)}\| \le 2\sqrt{\frac{\xi_{t-i-1}}{\lambda/2}}.$$

This yields the bound

$$\|y^{(t-1)} - y^{(t-2)}\|^2 \le 72\frac{\xi_{t-3}}{\lambda}.$$

All in all, we have obtained that

$$\tilde{D}_t(\alpha^*) - \tilde{D}_t(\alpha^{(t-1)}) \le \epsilon_{t-1} + \frac{36\kappa}{\lambda}\xi_{t-3}.$$

$\square$

Getting back to the proof of the second claim of Theorem 3, we have obtained that

$$\begin{aligned}
\frac{\tilde{D}_t(\alpha^*) - \tilde{D}_t(\alpha^{(t-1)})}{\frac{\eta}{2(1+\eta^{-2})}\xi_{t-1}} &\le \frac{\epsilon_{t-1}}{\frac{\eta}{2(1+\eta^{-2})}\xi_{t-1}} + \frac{36\kappa\xi_{t-3}}{\lambda\frac{\eta}{2(1+\eta^{-2})}\xi_{t-1}} \\
&\le (1-\eta/2)^{-1} + \frac{36\kappa 2(1+\eta^{-2})}{\lambda\eta}(1-\eta/2)^{-2} \\
&\le (1-\eta/2)^{-4}\left(1 + \frac{72\kappa(1+\eta^{-2})}{\lambda\eta}\right) \\
&\le (1-\eta/2)^{-2}\left(1 + 36\eta^{-5}\right),
\end{aligned}$$

where in the last inequality we used $\eta^{-2} - 1 = \frac{2\kappa}{\lambda}$, which implies that $\frac{2\kappa}{\lambda}(1+\eta^{-2}) \le \eta^{-4}$. Using $1 < \eta^{-5}$, $1 - \eta/2 \ge 0.5$, and taking log to both sides, we get that

$$\log\left(\frac{\tilde{D}_t(\alpha^*) - \tilde{D}_t(\alpha^{(t-1)})}{\frac{\eta}{2(1+\eta^{-2})}\xi_{t-1}}\right) \le 2\log(2) + \log(37) - 5\log(\eta) \le 7 + 2.5\log\left(\frac{R^2}{\lambda\gamma n}\right).$$

All in all, we have shown that the average runtime required by Prox-SDCA $(\tilde{P}_t, \frac{\eta}{2(1+\eta^{-2})}\xi_{t-1}, \alpha^{(t-1)})$ is upper bounded by

$$O\left(d\, n \log\left(\frac{R^2}{\lambda\gamma n}\right)\right),$$

which concludes the proof of the second claim of Theorem 3.

5 Applications

In this section we specify our algorithmic framework to several popular machine learning applications. In Sect. 5.1 we start by describing several loss functions and deriving their conjugate. In Sect. 5.2 we describe several regularization functions. Finally, in the rest of the subsections we specify our algorithm for Ridge regression, SVM, Lasso, logistic regression, and multiclass prediction.

5.1 Loss functions

Squared loss $\phi(a) = \frac{1}{2}(a-y)^2$ for some $y \in \mathbb{R}$. The conjugate function is

$$\phi^*(b) = \max_a ab - \frac{1}{2}(a-y)^2 = \frac{1}{2}b^2 + yb$$

Logistic loss $\phi(a) = \log(1+e^a)$. The derivative is $\phi'(a) = 1/(1+e^{-a})$ and the second derivative is $\phi''(a) = \frac{1}{(1+e^{-a})(1+e^a)} \in [0, 1/4]$, from which it follows that ϕ is $(1/4)$-smooth. The conjugate function is

$$\phi^*(b) = \max_a ab - \log(1+e^a) = \begin{cases} b\log(b) + (1-b)\log(1-b) & \text{if } b \in [0,1] \\ \infty & \text{otherwise} \end{cases}$$

Hinge loss $\phi(a) = [1-a]_+ := \max\{0, 1-a\}$. The conjugate function is

$$\phi^*(b) = \max_a ab - \max\{0, 1-a\} = \begin{cases} b & \text{if } b \in [-1,0] \\ \infty & \text{otherwise} \end{cases}$$

Smooth hinge loss This loss is obtained by smoothing the hinge-loss using the technique described in Lemma 2. This loss is parameterized by a scalar $\gamma > 0$ and is defined as:

$$\tilde{\phi}_\gamma(a) = \begin{cases} 0 & a \geq 1 \\ 1 - a - \gamma/2 & a \leq 1-\gamma \\ \frac{1}{2\gamma}(1-a)^2 & \text{o.w.} \end{cases} \tag{9}$$

The conjugate function is

$$\tilde{\phi}^*_\gamma(b) = \begin{cases} b + \frac{\gamma}{2}b^2 & \text{if } b \in [-1,0] \\ \infty & \text{otherwise} \end{cases}$$

It follows that $\tilde{\phi}^*_\gamma$ is γ strongly convex and $\tilde{\phi}$ is $(1/\gamma)$-smooth. In addition, if ϕ is the vanilla hinge-loss, we have for every a that

$$\phi(a) - \gamma/2 \leq \tilde{\phi}(a) \leq \phi(a) .$$

Remark 3 Usually, the hinge-loss is used as a convex surrogate for the zero-one loss (see for example [23], section 12.3). The experiments reported in [26] suggests that smoothing the hinge-loss has a very mild effect on the test zero-one error, hence, in practical applications, one can use large values of γ such as $\gamma = 1$.

Max-of-hinge The max-of-hinge loss function is a function from $\mathbb{R}^k$ to $\mathbb{R}$, which is defined as:

$$\phi(a) = \max_j \, [c_j + a_j]_+,$$

for some $c \in \mathbb{R}^k$. This loss function is useful for multiclass prediction problems.

To calculate the conjugate of ϕ, let

$$S = \{\beta \in \mathbb{R}^k_+ : \|\beta\|_1 \le 1\} \tag{10}$$

and note that we can write ϕ as

$$\phi(a) = \max_{\beta \in S} \sum_j \beta_j (c_j + a_j) \, .$$

Hence, the conjugate of ϕ is

$$\begin{aligned}
\phi^*(b) &= \max_a \left[a^\top b - \phi(a) \right] = \max_a \min_{\beta \in S} \left[a^\top b - \sum_j \beta_j (c_j + a_j) \right] \\
&= \min_{\beta \in S} \max_a \left[a^\top b - \sum_j \beta_j (c_j + a_j) \right] \\
&= \min_{\beta \in S} \left[- \sum_j \beta_j c_j + \sum_j \max_{a_j} a_j (b_j - \beta_j) \right] .
\end{aligned}$$

Each inner maximization over a_j would be ∞ unless $\beta_j = b_j$. Therefore,

$$\phi^*(b) = \begin{cases} -c^\top b & \text{if } b \in S \\ \infty & \text{otherwise} \end{cases} \tag{11}$$

Smooth max-of-hinge This loss obtained by smoothing the max-of-hinge loss using the technique described in Lemma 2. This loss is parameterized by a scalar $\gamma > 0$. We start by adding regularization to the conjugate of the max-of-hinge given in (11) and obtain

$$\tilde{\phi}^*_\gamma(b) = \begin{cases} \frac{\gamma}{2}\|b\|^2 - c^\top b & \text{if } b \in S \\ \infty & \text{otherwise} \end{cases} \tag{12}$$

Taking the conjugate of the conjugate we obtain

$$\begin{aligned}\tilde{\phi}_\gamma(a) &= \max_b b^\top a - \tilde{\phi}^*_\gamma(b)\\ &= \max_{b\in S} b^\top(a+c) - \frac{\gamma}{2}\|b\|^2\\ &= \frac{\gamma}{2}\|(a+c)/\gamma\|^2 - \frac{\gamma}{2}\min_{b\in S}\|b-(a+c)/\gamma\|^2\end{aligned} \tag{13}$$

While we do not have a closed form solution for the minimization problem over b in the definition of $\tilde{\phi}_\gamma$ above, this is a problem of projecting onto the intersection of the L_1 ball and the positive orthant, and can be solved efficiently using the following procedure, adapted from [9].

Project(μ)

Goal: solve $\mathrm{argmin}_b\ \|b-\mu\|^2$ s.t. $b \in \mathbb{R}^k_+$, $\|b\|_1 \le 1$
Let: $\forall i,\ \tilde{\mu}_i = \max\{0, \mu_i\}$
If: $\|\tilde{\mu}\|_1 \le 1$ stop and return $b = \tilde{\mu}$
Sort: let $i_1, \ldots, i_k$ be s.t. $\mu_{i_1} \ge \mu_{i_2} \ge \ldots \ge \mu_{i_k}$
Find: $j^* = \max\left\{j : j\,\tilde{\mu}_{i_j} + 1 - \sum_{r=1}^{j}\tilde{\mu}_{i_r} > 0\right\}$
Define: $\theta = -1 + \sum_{r=1}^{j^*}\tilde{\mu}_{i_r}$
Return: b s.t. $\forall i,\ b_i = \max\{\mu_i - \theta/j^*, 0\}$

It also holds that $\nabla\tilde{\phi}_\gamma(a) = \mathrm{argmin}_{b\in S}\|b-(a+c)/\gamma\|^2$, and therefore the gradient can also be calculated using the above projection procedure.

Note that if ϕ being the max-of-hinge loss, then $\phi^*(b) + \gamma/2 \ge \tilde{\phi}^*_\gamma(b) \ge \phi^*(b)$ and hence $\phi(a) - \gamma/2 \le \tilde{\phi}_\gamma(a) \le \phi(a)$.

Observe that all negative elements of $a + c$ does not contribute to $\tilde{\phi}_\gamma$. This immediately implies that if $\phi(a) = 0$ then we also have $\tilde{\phi}_\gamma(a) = 0$.

Soft-max-of-hinge loss function Another approach to smooth the max-of-hinge loss function is by using soft-max instead of max. The resulting soft-max-of-hinge loss function is defined as

$$\phi_\gamma(a) = \gamma\log\left(1 + \sum_{i=1}^{k} e^{(c_i+a_i)/\gamma}\right), \tag{14}$$

where $\gamma > 0$ is a parameter. We have

$$\max_i [c_i + a_i]_+ \le \phi_\gamma(a) \le \max_i [c_i + a_i]_+ + \gamma\ \log(k+1).$$

The j'th element of the gradient of ϕ is

$$\nabla_j\phi_\gamma(a) = \frac{e^{(c_j+a_j)/\gamma}}{1 + \sum_{i=1}^{k} e^{(c_i+a_i)/\gamma}}.$$

By the definition of the conjugate we have $\phi_\gamma^*(b) = \max_a a^\top b - \phi_\gamma(a)$. The vector a that maximizes the above must satisfy

$$\forall j, b_j = \frac{e^{(c_j+a_j)/\gamma}}{1+\sum_{i=1}^k e^{(c_i+a_i)/\gamma}}.$$

This can be satisfied only if $b_j \geq 0$ for all j and $\sum_j b_j \leq 1$. That is, $b \in S$. Denote $Z = \sum_{i=1}^k e^{(c_i+a_i)/\gamma}$ and note that

$$(1+Z)\|b\|_1 = Z \quad \Rightarrow \quad Z = \frac{\|b\|_1}{1-\|b\|_1} \quad \Rightarrow \quad 1+Z = \frac{1}{1-\|b\|_1}.$$

It follows that

$$a_j = \gamma(\log(b_j) + \log(1+Z)) - c_j = \gamma(\log(b_j) - \log(1-\|b\|_1)) - c_j$$

which yields

$$\begin{aligned} \phi_\gamma^*(b) &= \sum_j \left(\gamma(\log(b_j) - \log(1-\|b\|_1)) - c_j\right) b_j + \gamma \log(1-\|b\|_1) \\ &= -c^\top b + \gamma \left((1-\|b\|_1)\log(1-\|b\|_1) + \sum_j b_j \log(b_j) \right). \end{aligned}$$

Finally, if $b \notin S$ then the gradient of $a^\top b - \phi_\gamma(a)$ does not vanish anywhere, which means that $\phi_\gamma^*(b) = \infty$. All in all, we obtain

$$\phi_\gamma^*(b) = \begin{cases} -c^\top b + \gamma \left((1-\|b\|_1)\log(1-\|b\|_1) + \sum_j b_j \log(b_j) \right) & \text{if } b \in S \\ \infty & \text{otherwise} \end{cases} \tag{15}$$

Since the entropic function, $\sum_j b_j \log(b_j)$ is 1-strongly convex over S with respect to the L_1 norm, we obtain that ϕ_γ^* is γ-strongly convex with respect to the L_1 norm, from which it follows that ϕ_γ is $(1/\gamma)$-smooth with respect to the L_∞ norm.

5.2 Regularizers

L_2 **regularization** The simplest regularization is the squared L_2 regularization

$$g(w) = \frac{1}{2}\|w\|_2^2.$$

This is a 1-strongly convex regularization function whose conjugate is

$$g^*(\theta) = \frac{1}{2}\|\theta\|_2^2.$$

We also have

$$\nabla g^*(\theta) = \theta.$$

For our acceleration procedure, we also use the L_2 regularization plus a linear term, namely,

$$g(w) = \frac{1}{2}\|w\|^2 - w^\top z,$$

for some vector z. The conjugate of this function is

$$g^*(\theta) = \max_w \left[w^\top (\theta + z) - \frac{1}{2}\|w\|^2 \right] = \frac{1}{2}\|\theta + z\|^2.$$

We also have

$$\nabla g^*(\theta) = \theta + z.$$

L_1 **regularization** Another popular regularization we consider is the L_1 regularization,

$$f(w) = \sigma \, \|w\|_1.$$

This is not a strongly convex regularizer and therefore we will add a slight L_2 regularization to it and define the L_1-L_2 regularization as

$$g(w) = \frac{1}{2}\|w\|_2^2 + \sigma' \, \|w\|_1, \tag{16}$$

where $\sigma' = \frac{\sigma}{\lambda}$ for some small λ. Note that

$$\lambda g(w) = \frac{\lambda}{2}\|w\|_2^2 + \sigma \|w\|_1,$$

so if λ is small enough (as will be formalized later) we obtain that $\lambda g(w) \approx \sigma \|w\|_1$.

The conjugate of g is

$$g^*(v) = \max_w \left[w^\top v - \frac{1}{2}\|w\|_2^2 - \sigma' \|w\|_1 \right].$$

The maximizer is also $\nabla g^*(v)$ and we now show how to calculate it. We have

$$\begin{aligned}\nabla g^*(v) &= \underset{w}{\operatorname{argmax}} \left[w^\top v - \frac{1}{2}\|w\|_2^2 - \sigma' \|w\|_1 \right] \\ &= \underset{w}{\operatorname{argmin}} \left[\frac{1}{2}\|w - v\|_2^2 + \sigma' \|w\|_1 \right]\end{aligned}$$

A sub-gradient of the objective of the optimization problem above is of the form $w - v + \sigma' z = 0$, where z is a vector with $z_i = \operatorname{sign}(w_i)$, where if $w_i = 0$ then $z_i \in [-1, 1]$. Therefore, if w is an optimal solution then for all i, either $w_i = 0$ or $w_i = v_i - \sigma' \operatorname{sign}(w_i)$. Furthermore, it is easy to verify that if w is an optimal solution then for all i, if $w_i \neq 0$ then the sign of w_i must be the sign of v_i. Therefore, whenever $w_i \neq 0$ we have that $w_i = v_i - \sigma' \operatorname{sign}(v_i)$. It follows that in that case we must have $|v_i| > \sigma'$. And, the other direction is also true, namely, if $|v_i| > \sigma'$ then setting $w_i = v_i - \sigma' \operatorname{sign}(v_i)$ leads to an objective value whose i'th component is

$$\frac{1}{2}\left(\sigma'\right)^2 + \sigma'(|v_i| - \sigma') \leq \frac{1}{2}|v_i|^2,$$

where the right-hand side is the i'th component of the objective value we will obtain by setting $w_i = 0$. This leads to the conclusion that

$$\nabla_i g^*(v) = \operatorname{sign}(v_i)\left[|v_i| - \sigma'\right]_+ = \begin{cases} v_i - \sigma' \operatorname{sign}(v_i) & \text{if } |v_i| > \sigma' \\ 0 & \text{o.w.} \end{cases}$$

It follows that

$$\begin{aligned}g^*(v) &= \sum_i \operatorname{sign}(v_i)\left[|v_i| - \sigma'\right]_+ v_i - \frac{1}{2}\sum_i \left(\left[|v_i| - \sigma'\right]_+\right)^2 - \sigma' \sum_i \left[|v_i| - \sigma'\right]_+ \\ &= \sum_i \left[|v_i| - \sigma'\right]_+ \left(|v_i| - \sigma' - \frac{1}{2}\left[|v_i| - \sigma'\right]_+\right) = \frac{1}{2}\sum_i \left(\left[|v_i| - \sigma'\right]_+\right)^2.\end{aligned}$$

Another regularization function we'll use in the accelerated procedure is

$$g(w) = \frac{1}{2}\|w\|_2^2 + \sigma' \|w\|_1 - z^\top w\ . \tag{17}$$

The conjugate function is

$$g^*(v) = \frac{1}{2}\sum_i \left(\left[|v_i + z_i| - \sigma'\right]_+\right)^2,$$

and its gradient is

$$\nabla_i g^*(v) = \operatorname{sign}(v_i + z_i)\left[|v_i + z_i| - \sigma'\right]_+$$

5.3 Ridge regression

In ridge regression, we minimize the squared loss with L_2 regularization. That is, $g(w) = \frac{1}{2}\|w\|^2$ and for every i we have that $x_i \in \mathbb{R}^d$ and $\phi_i(a) = \frac{1}{2}(a - y_i)^2$ for some $y_i \in \mathbb{R}$. The primal problem is therefore

$$P(w) = \frac{1}{2n}\sum_{i=1}^{n}(x_i^\top w - y_i)^2 + \frac{\lambda}{2}\|w\|^2.$$

Below we specify Prox-SDCA for ridge regression. We use Option I since it is possible to derive a closed form solution to the maximization of the dual with respect to $\Delta\alpha_i$. Indeed, since $-\phi_i^*(-b) = -\frac{1}{2}b^2 + y_i b$ we have that the maximization problem is

$$\begin{aligned}
\Delta\alpha_i &= \operatorname*{argmax}_b -\frac{1}{2}(\alpha_i^{(t+1)} + b)^2 + y_i(\alpha_i^{(t+1)} + b) - w^{(t-1)\top} x_i b - \frac{b^2\|x_i\|^2}{2\lambda n} \\
&= \operatorname*{argmax}_b -\frac{1}{a}\left(1 + \frac{\|x_i\|^2}{2\lambda n}\right) b^2 - \left(\alpha_i^{(t+1)} + w^{(t-1)\top} x_i - y_i\right) b \\
&= -\frac{\alpha_i^{(t+1)} + w^{(t-1)\top} x_i - y_i}{1 + \frac{\|x_i\|^2}{2\lambda n}}.
\end{aligned}$$

Applying the above update and using some additional tricks to improve the running time we obtain the following procedure.

Prox-SDCA($(x_i, y_i)_{i=1}^n, \epsilon, \alpha^{(0)}, z$) **for solving ridge regression**

Goal: Minimize $P(w) = \frac{1}{2n}\sum_{i=1}^n (x_i^\top w - y_i)^2 + \lambda\left(\frac{1}{2}\|w\|^2 - w^\top z\right)$
Initialize $v^{(0)} = \frac{1}{\lambda n}\sum_{i=1}^n \alpha_i^{(0)} x_i$, $\forall i,\ \tilde{y}_i = y_i - x_i^\top z$
Iterate: for $t = 1, 2, \ldots$
  Randomly pick i
  $\Delta\alpha_i = -\frac{\alpha_i^{(t-1)} + v^{(t-1)\top} x_i - \tilde{y}_i}{1 + \frac{\|x_i\|^2}{2\lambda n}}$
  $\alpha_i^{(t)} \leftarrow \alpha_i^{(t-1)} + \Delta\alpha_i$ and for $j \neq i$, $\alpha_j^{(t)} \leftarrow \alpha_j^{(t-1)}$
  $v^{(t)} \leftarrow v^{(t-1)} + \frac{\Delta\alpha_i}{\lambda n} x_i$
Stopping condition:
  Let $w^{(t)} = v^{(t)} + z$
  Stop if $\frac{1}{2n}\sum_{i=1}^n \left((x_i^\top w^{(t)} - y_i)^2 + (\alpha_i^{(t)} + y_i)^2 - y_i^2\right) + \lambda w^{(t)\top} v^{(t)} \le \epsilon$

The runtime of Prox-SDCA for ridge regression becomes

$$\tilde{O}\left(d\left(n + \frac{R^2}{\lambda}\right)\right),$$

where $R = \max_i \|x_i\|$. This matches the recent results of [16,26]. If $R^2/\lambda \gg n$ we can apply the accelerated procedure and obtain the improved runtime

$$\tilde{O}\left(d\sqrt{\frac{nR^2}{\lambda}}\right) .$$

5.4 Logistic regression

In logistic regression, we minimize the logistic loss with L_2 regularization. That is, $g(w) = \frac{1}{2}\|w\|^2$ and for every i we have that $x_i \in \mathbb{R}^d$ and $\phi_i(a) = log(1 + e^a)$. The primal problem is therefore[3]

$$P(w) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{x_i^\top w}) + \frac{\lambda}{2}\|w\|^2.$$

The dual problem is

$$D(\alpha) = \frac{1}{n}\sum_{i=1}^{n}(\alpha_i \log(-\alpha_i) - (1 + \alpha_i)\log(1 + \alpha_i)) - \frac{\lambda}{2}\|v(\alpha)\|^2,$$

and the dual constraints are $\alpha \in [-1, 0]^n$.

Below we specify Prox-SDCA for logistic regression using Option III.

Prox-SDCA($(x_i)_{i=1}^n, \epsilon, \alpha^{(0)}, z$) for logistic regression

Goal: Minimize $P(w) = \frac{1}{n}\sum_{i=1}^n \log(1 + e^{x_i^\top w}) + \lambda\left(\frac{1}{2}\|w\|^2 - w^\top z\right)$
Initialize $v^{(0)} = \frac{1}{\lambda n}\sum_{i=1}^n \alpha_i^{(0)} x_i$, and $\forall i,\quad p_i = x_i^\top z$
Define: $\phi^*(b) = b\log(b) + (1 - b)\log(1 - b)$
Iterate: for $t = 1, 2, \ldots$
 Randomly pick i
 $p = x_i^\top w^{(t-1)}$
 $q = -1/(1 + e^{-p}) - \alpha_i^{(t-1)}$
 $s = \min\left(1, \frac{\log(1+e^p)+\phi^*(-\alpha_i^{(t-1)})+p\alpha_i^{(t-1)}+2q^2}{q^2(4+\frac{1}{\lambda n}\|x_i\|^2)}\right)$
 $\Delta\alpha_i = sq$
 $\alpha_i^{(t)} = \alpha_i^{(t-1)} + \Delta\alpha_i$ and for $j \neq i$, $\alpha_j^{(t)} = \alpha_j^{(t-1)}$
 $v^{(t)} = v^{(t-1)} + \frac{\Delta\alpha_i}{\lambda n} x_i$
Stopping condition:
 let $w^{(t)} = v^{(t)} + z$
 Stop if $\frac{1}{n}\sum_{i=1}^n \left(\log(1 + e^{x_i^\top w^{(t)}}) + \phi^*(-\alpha_i^{(t-1)})\right) + \lambda w^{(t)\top} v^{(t)} \leq \epsilon$

[3] Usually, the training data comes with labels, $y_i \in \{\pm 1\}$, and the loss function becomes $\log(1+e^{-y_i x_i^\top w})$. However, we can easily get rid of the labels by re-defining $x_i \leftarrow -y_i x_i$.

The runtime analysis is similar to the analysis for ridge regression.

5.5 Lasso

In the Lasso problem, the loss function is the squared loss but the regularization function is L_1. That is, we need to solve the problem:

$$\min_w \left[\frac{1}{2n} \sum_{i=1}^n (x_i^\top w - y_i)^2 + \sigma \|w\|_1 \right], \tag{18}$$

with a positive regularization parameter $\sigma \in \mathbb{R}_+$.

Let $\bar{y} = \frac{1}{2n} \sum_{i=1}^n y_i^2$, and let $\bar{w}$ be an optimal solution of (18). Then, the objective at $\bar{w}$ is at most the objective at $w = 0$, which yields

$$\sigma \|\bar{w}\|_1 \le \bar{y} \Rightarrow \|\bar{w}\|_2 \le \|\bar{w}\|_1 \le \frac{\bar{y}}{\sigma}.$$

Consider the optimization problem

$$\min_w P(w) \quad \text{where} \quad P(w) = \frac{1}{2n} \sum_{i=1}^n (x_i^\top w - y_i)^2 + \lambda \left(\frac{1}{2} \|w\|_2^2 + \frac{\sigma}{\lambda} \|w\|_1 \right), \tag{19}$$

for some $\lambda > 0$. This problem fits into our framework, since now the regularizer is strongly convex. Furthermore, if w^* is an $(\epsilon/2)$-accurate solution to the problem in (19), then $P(w^*) \le P(\bar{w}) + \epsilon/2$ which yields

$$\begin{aligned} &\left[\frac{1}{2n} \sum_{i=1}^n (x_i^\top w^* - y_i)^2 + \sigma \|w^*\|_1 \right] \\ &\quad \le \left[\frac{1}{2n} \sum_{i=1}^n (x_i^\top \bar{w} - y_i)^2 + \sigma \|\bar{w}\|_1 \right] + \frac{\lambda}{2} \|\bar{w}\|_2^2 + \epsilon/2. \end{aligned}$$

Since $\|\bar{w}\|_2^2 \le (\bar{y}/\sigma)^2$, we obtain that setting $\lambda = \epsilon(\sigma/\bar{y})^2$ guarantees that w^* is an ϵ accurate solution to the original problem given in (18).

In light of the above, from now on we focus on the problem given in (19). As in the case of ridge regression, we can apply Prox-SDCA with Option I. The resulting pseudo-code is given below. Applying the above update and using some additional tricks to improve the running time we obtain the following procedure.

Prox-SDCA($(x_i, y_i)_{i=1}^n, \epsilon, \alpha^{(0)}, z$ **) for solving** $L_1 - L_2$ **regression**

Goal: Minimize $P(w) = \frac{1}{2n}\sum_{i=1}^n (x_i^\top w - y_i)^2 + \lambda\left(\frac{1}{2}\|w\|^2 + \sigma'\|w\|_1 - w^\top z\right)$
Initialize $v^{(0)} = \frac{1}{\lambda n}\sum_{i=1}^n \alpha_i^{(0)} x_i$, and $\forall j,\ w_j^{(0)} = \mathrm{sign}(v_j^{(0)} + z_j)[|v_j^{(0)} + z_j| - \sigma']_+$
Iterate: for $t = 1, 2, \ldots$
Randomly pick i
$\Delta\alpha_i = -\frac{\alpha_i^{(t-1)} + w^{(t-1)\top} x_i - y_i}{1 + \frac{\|x_i\|^2}{2\lambda n}}$
$\alpha_i^{(t)} = \alpha_i^{(t-1)} + \Delta\alpha_i$ and for $j \neq i$, $\alpha_j^{(t)} = \alpha_j^{(t-1)}$
$v^{(t)} = v^{(t-1)} + \frac{\Delta\alpha_i}{\lambda n} x_i$
$\forall j,\ w_j^{(t)} = \mathrm{sign}(v_j^{(t)} + z_j)[|v_j^{(t)} + z_j| - \sigma']_+$
Stopping condition:
Stop if $\frac{1}{2n}\sum_{i=1}^n \left((x_i^\top w^{(t)} - y_i)^2 - 2y_i\alpha_i^{(t)} + (\alpha_i^{(t)})^2\right) + \lambda w^{(t)\top} v^{(t)} \leq \epsilon$

Let us now discuss the runtime of the resulting method. Denote $R = \max_i \|x_i\|$ and for simplicity, assume that $\bar{y} = O(1)$. Choosing $\lambda = \epsilon(\sigma/\bar{y})^2$, the runtime of our method becomes

$$\tilde{O}\left(d\left(n + \min\left\{\frac{R^2}{\epsilon\,\sigma^2}, \sqrt{\frac{n R^2}{\epsilon\,\sigma^2}}\right\}\right)\right).$$

It is also convenient to write the bound in terms of $B = \|\bar{w}\|_2$, where, as before, $\bar{w}$ is the optimal solution of the L_1 regularized problem. With this parameterization, we can set $\lambda = \epsilon/B^2$ and the runtime becomes

$$\tilde{O}\left(d\left(n + \min\left\{\frac{R^2 B^2}{\epsilon}, \sqrt{\frac{n\,R^2 B^2}{\epsilon}}\right\}\right)\right).$$

The runtime of standard SGD is $O(dR^2B^2/\epsilon^2)$ even in the case of smooth loss functions such as the squared loss. Several variants of SGD, that leads to sparser intermediate solutions, have been proposed (e.g. [8,10,15,22,30]). However, all of these variants share the runtime of $O(dR^2B^2/\epsilon^2)$, which is much slower than our runtime when ϵ is small.

Another relevant approach is the FISTA algorithm of [2]. The shrinkage operator of FISTA is the same as the gradient of g^* used in our approach. It is a batch algorithm using Nesterov's accelerated gradient technique. For the squared loss function, the runtime of FISTA is

$$O\left(d\,n\sqrt{\frac{R^2B^2}{\epsilon}}\right).$$

This bound is worst than our bound by a factor of at least $\sqrt{n}$.

Another approach to solving (18) is stochastic coordinate descent over the primal problem. [22] showed that the runtime of this approach is

$$O\left(\frac{dnB^2}{\epsilon}\right),$$

under the assumption that $\|x_i\|_\infty \le 1$ for all i. Similar results can also be found in [20]. Moreover, a recent result in [11] shows that under similar assumptions, acceleration can be used to improve the runtime to

$$O\left(\frac{dnB}{\sqrt{\epsilon}}\right).$$

For our method, the runtime depends on $R^2 = \max_i \|x_i\|_2^2$. If $R^2 = O(1)$ then the runtime of our method is better than that of [22] and [11]. In the general case, if $\max_i \|x_i\|_\infty \le 1$ then $R^2 \le d$, which yields the runtime of

$$\tilde{O}\left(d\left(n + \min\left\{\frac{dB^2}{\epsilon}, \sqrt{\frac{n\,dB^2}{\epsilon}}\right\}\right)\right).$$

This is better than [22] and [11] when d is small but worse when d is large.

5.6 Linear SVM

Support Vector Machines (SVM) is an algorithm for learning a linear classifier. Linear SVM (i.e., SVM with linear kernels) amounts to minimizing the objective

$$P(w) = \frac{1}{n}\sum_{i=1}^{n}[1 - x_i^\top w]_+ + \frac{\lambda}{2}\|w\|^2,$$

where $[a]_+ = \max\{0, a\}$, and for every i, $x_i \in \mathbb{R}^d$. This can be cast as the objective given in (1) by letting the regularization be $g(w) = \frac{1}{2}\|w\|_2^2$, and for every i, $\phi_i(a) = [1 - a]_+$, is the hinge-loss.

Let $R = \max_i \|x_i\|_2$. SGD enjoys the rate of $O\left(\frac{1}{\lambda\epsilon}\right)$. Many software packages apply SDCA and obtain the rate $\tilde{O}\left(n + \frac{1}{\lambda\epsilon}\right)$. We now show how our accelerated proximal SDCA enjoys the rate $\tilde{O}\left(n + \sqrt{\frac{n}{\lambda\epsilon}}\right)$. This is significantly better than the rate of SGD when $\lambda\epsilon < 1/n$. We note that a default setting for λ, which often works well in practice, is $\lambda = 1/n$. In this case, $\lambda\epsilon = \epsilon/n \ll 1/n$.

Our first step is to smooth the hinge-loss. Let $\gamma = \epsilon$ and consider the smooth hinge-loss as defined in (9). Recall that the smooth hinge-loss satisfies

$$\forall a, \phi(a) - \gamma/2 \le \tilde{\phi}(a) \le \phi(a).$$

Let $\tilde{P}$ be the SVM objective while replacing the hinge-loss with the smooth hinge-loss. Therefore, for every w' and w,

$$P(w') - P(w) \le \tilde{P}(w') - \tilde{P}(w) + \gamma/2.$$

It follows that if w' is an $(\epsilon/2)$-optimal solution for $\tilde{P}$, then it is ϵ-optimal solution for P.

For the smoothed hinge loss, the optimization problem given in Option I of Prox-SDCA has a closed form solution and we obtain the following procedure:

Prox-SDCA($(x_1, \ldots, x_n), \epsilon, \alpha^{(0)}, z$) for solving SVM (with smooth hinge-loss as in (9))

Define: $\tilde{\phi}_\gamma$ as in (9)
Goal: Minimize $P(w) = \frac{1}{n}\sum_{i=1}^n \tilde{\phi}_\gamma(x_i^\top w) + \lambda\left(\frac{1}{2}\|w\|^2 - w^\top z\right)$
Initialize $w^{(0)} = z + \frac{1}{\lambda n}\sum_{i=1}^n \alpha_i^{(0)} x_i$
Iterate: for $t = 1, 2, \ldots$
Randomly pick i
$\Delta\alpha_i = \max\left(-\alpha_i^{(t-1)}\,,\ \min\left(1 - \alpha_i^{(t-1)}\,,\ \frac{1 - x_i^\top w^{(t-1)} - \gamma\,\alpha_i^{(t-1)}}{\|x_i\|^2/(\lambda n) + \gamma}\right)\right)$
$\alpha_i^{(t)} \leftarrow \alpha_i^{(t-1)} + \Delta\alpha_i$ and for $j \ne i$, $\alpha_j^{(t)} \leftarrow \alpha_j^{(t-1)}$
$w^{(t)} \leftarrow w^{(t-1)} + \frac{\Delta\alpha_i}{\lambda n} x_i$
Stopping condition:
Stop if $\frac{1}{n}\sum_{i=1}^n \left(\tilde{\phi}_\gamma(x_i^\top w^{(t)}) - \alpha_i^{(t)} + \frac{\gamma}{2}(\alpha_i^{(t)})^2\right) + \lambda w^{(t)^\top}(w^{(t)} - z) \le \epsilon$

Denote $R = \max_i \|x_i\|$. Then, the runtime of the resulting method is

$$\tilde{O}\left(d\left(n + \min\left\{\frac{R^2}{\gamma\,\lambda}, \sqrt{\frac{n R^2}{\gamma\,\lambda}}\right\}\right)\right).$$

In particular, choosing $\gamma = \epsilon$ we obtain a solution to the original SVM problem in runtime of

$$\tilde{O}\left(d\left(n + \min\left\{\frac{R^2}{\epsilon\,\lambda}, \sqrt{\frac{n R^2}{\epsilon\,\lambda}}\right\}\right)\right).$$

As mentioned before, this is better than SGD when $\frac{1}{\lambda\epsilon} \gg n$.

5.7 Multiclass SVM

Next we consider Multiclass SVM using the construction described in Crammer and Singer [5]. Each example consists of an instance vector $x_i \in \mathbb{R}^d$ and a label $y_i \in \{1, \ldots, k\}$. The goal is to learn a matrix $W \in \mathbb{R}^{d,k}$ such that $W^\top x_i$ is a k'th dimensional vector of scores for the different classes. The prediction is the coordinate of $W^\top x_i$ of maximal value. The loss function is

$$\max_{j\neq y_i}(1+(W^\top x_i)_j-(W^\top x_i)_{y_i}).$$

This can be written as $\phi((W^\top x_i)-(W^\top x_i)_{y_i})$ where

$$\phi_i(a)=\max_j[c_{i,j}+a_j]_+,$$

with c_i being the all ones vector except 0 in the y_i coordinate.

We can model this in our framework as follows. Given a matrix M let $\mathrm{vec}(M)$ be the column vector obtained by concatenating the columns of M. Let e_j be the all zeros vector except 1 in the j'th coordinate. For every i, let $c_i=\mathbf{1}-e_{y_i}$ and let $X_i\in\mathbb{R}^{dk,k}$ be the matrix whose j'th column is $\mathrm{vec}(x_i(e_j-e_{y_i})^\top)$. Then,

$$X_i^\top \mathrm{vec}(W)=W^\top x_i-(W^\top x_i)_{y_i}\ .$$

Therefore, the optimization problem of multiclass SVM becomes:

$$\min_{w\in\mathbb{R}^{dk}} P(w)\quad\text{where}\quad P(w)=\frac{1}{n}\sum_{i=1}^n\phi_i(X_i^\top w)+\frac{\lambda}{2}\|w\|^2.$$

As in the case of SVM, we will use the smooth version of the max-of-hinge loss function as described in (13). If we set the smoothness parameter γ to be ϵ then an $(\epsilon/2)$-accurate solution to the problem with the smooth loss is also an ϵ-accurate solution to the original problem with the non-smooth loss. Therefore, from now on we focus on the problem with the smooth max-of-hinge loss.

We specify Prox-SDCA for multiclass SVM using Option I. We will show that the optimization problem in Option I can be calculated efficiently by sorting a k dimensional vector. Such ideas were explored in [5] for the non-smooth max-of-hinge loss.

Let $\hat{w}=w-\frac{1}{\lambda n}X_i\alpha_i^{(t-1)}$. Then, the optimization problem over α_i can be written as

$$\operatorname*{argmax}_{\alpha_i:-\alpha_i\in S}\ (-c_i^\top-\hat{w}^\top X_i)\alpha_i-\frac{\gamma}{2}\|\alpha_i\|^2-\frac{1}{2\lambda n}\|X_i\alpha_i\|^2. \tag{20}$$

As shown before, if we organize $\hat{w}$ as a $d\times k$ matrix, denoted $\hat{W}$, we have that $X_i^\top\hat{w}=\hat{W}^\top x_i-(\hat{W}^\top x_i)_{y_i}$. We also have that

$$\begin{aligned}X_i\alpha_i&=\sum_j \mathrm{vec}\left(x_i(e_j-e_{y_i})^\top\right)\alpha_{i,j}=\mathrm{vec}\left(x_i\sum_j\alpha_{i,j}\left(e_j-e_{y_i}\right)^\top\right)\\&=\mathrm{vec}\left(x_i(\alpha_i-\|\alpha_i\|_1 e_{y_i})^\top\right).\end{aligned}$$

It follows that an optimal solution to (20) must set $\alpha_{i,y_i} = 0$ and we only need to optimize over the rest of the dual variables. This also yields,

$$\|X_i\alpha_i\|^2 = \|x_i\|^2\|\alpha_i\|_2^2 + \|x_i\|^2\|\alpha_i\|_1^2.$$

So, (20) becomes:

$$\operatorname*{argmax}_{\alpha_i : -\alpha_i \in S, \alpha_{i,y_i}=0} \left(-c_i^\top - \hat{w}^\top X_i\right)\alpha_i - \frac{\gamma}{2}\|\alpha_i\|_2^2 - \frac{\|x_i\|^2}{2\lambda n}\|\alpha_i\|_2^2 - \frac{\|x_i\|^2}{2\lambda n}\|\alpha_i\|_1^2. \tag{21}$$

This is equivalent to a problem of the form:

$$\operatorname*{argmin}_{a \in \mathbb{R}_+^{k-1}, \beta} \|a - \mu\|_2^2 + C\beta^2 \quad \text{s.t.} \quad \|a\|_1 = \beta \le 1, \tag{22}$$

where

$$\mu = \frac{c_i^\top + \hat{w}^\top X_i}{\gamma + \frac{\|x_i\|^2}{\lambda n}} \quad \text{and} \quad C = \frac{\frac{\|x_i\|^2}{\lambda n}}{\gamma + \frac{\|x_i\|^2}{\lambda n}} = \frac{1}{\frac{\gamma\lambda n}{\|x_i\|^2} + 1}.$$

The equivalence is in the sense that if (a, β) is a solution of (22) then we can set $\alpha_i = -a$.

Assume for simplicity that μ is sorted in a non-increasing order and that all of its elements are non-negative (otherwise, it is easy to verify that we can zero the negative elements of μ and sort the non-negative, without affecting the solution). Let $\bar{\mu}$ be the cumulative sum of μ, that is, for every j, let $\bar{\mu}_j = \sum_{r=1}^{j} \mu_r$. For every j, let $z_j = \bar{\mu}_j - j\mu_j$. Since μ is sorted we have that

$$z_{j+1} = \sum_{r=1}^{j+1} \mu_r - (j+1)\mu_{j+1} = \sum_{r=1}^{j} \mu_r - j\mu_{j+1} \ge \sum_{r=1}^{j} \mu_r - j\mu_j = z_j.$$

Note also that $z_1 = 0$ and that $z_k = \bar{\mu}_k = \|\mu\|_1$ (since the coordinate of μ that corresponds to y_i is zero). By the properties of projection onto the simplex (see [9]), for every $z \in (z_j, z_{j+1})$ we have that the projection of μ onto the set $\{b \in \mathbb{R}_+^k : \|b\|_1 = z\}$ is of the form $a_r = \max\{0, \mu_r - \theta/j\}$ where $\theta = (-z + \bar{\mu}_j)/j$. Therefore, the objective becomes (ignoring constants that do not depend on z),

$$j\theta^2 + Cz^2 = (-z + \bar{\mu}_j)^2/j + Cz^2.$$

The first order condition for minimality w.r.t. z is

$$-(-z + \bar{\mu}_j)/j + Cz = 0 \Rightarrow z = \frac{\bar{\mu}_j}{1 + jC}.$$

If this value of z is in (z_j, z_{j+1}), then it is the optimal z and we're done. Otherwise, the optimum should be either $z = 0$ (which yields $\alpha = 0$ as well) or $z = 1$.

$a = \mathrm{OptimizeDual}(\mu, C)$

Solve the optimization problem given in (22)
Initialize: $\forall i,\ \hat{\mu}_i = \max\{0, \mu_i\}$, and sort $\hat{\mu}$ s.t. $\hat{\mu}_1 \geq \hat{\mu}_2 \geq \ldots \geq \hat{\mu}_k$
Let: $\bar{\mu}$ be s.t. $\bar{\mu}_j = \sum_{i=1}^{j} \hat{\mu}_i$
Let: z be s.t. $z_j = \min\{\bar{\mu}_j - j\hat{\mu}_j, 1\}$ and $z_{k+1} = 1$
If: $\exists j$ s.t. $\frac{\bar{\mu}_j}{1+jC} \in [z_j, z_{j+1}]$
  return a s.t. $\forall i,\ a_i = \max\left\{0, \mu_i - \left(-\frac{\bar{\mu}_j}{1+jC} + \bar{\mu}_j\right)/j\right\}$
Else:
  Let j be the minimal index s.t. $z_j = 1$
  set a s.t. $\forall i,\ a_i = \max\{0, \mu_i - (-z_j + \bar{\mu}_j)/j\}$
  If: $\|a - \mu\|^2 + C \leq \|\mu\|^2$
    return a
  Else:
    return $(0, \ldots, 0)$

The resulting pseudo-codes for Prox-SDCA is given below. We specify the procedure while referring to W as a matrix, because it is the more natural representation. For convenience of the code, we also maintain in α_{i,y_i} the value of $-\sum_{j \neq y_i} \alpha_{i,j}$ (instead of the optimal value of 0).

Prox-SDCA($(x_1, y_1)_{i=1}^n, \epsilon, \alpha, Z$) for solving Multiclass SVM (with smooth hinge-loss as in (13))

Define: $\tilde{\phi}_\gamma$ as in (13)
Goal: Minimize
$P(W) = \frac{1}{n}\sum_{i=1}^n \tilde{\phi}_\gamma((W^\top x_i) - (W^\top x_i)_{y_i}) + \lambda\left(\frac{1}{2}\mathrm{vec}(W)^\top \mathrm{vec}(W) - \mathrm{vec}(W)^\top \mathrm{vec}(Z)\right)$
Initialize $W = Z + \frac{1}{\lambda n}\sum_{i=1}^n x_i \alpha_i^\top$
Iterate: for $t = 1, 2, \ldots$
  Randomly pick i
  $\hat{W} = W - \frac{1}{\lambda n} x_i \alpha_i^\top$
  $p = x_i^\top \hat{W},\quad p = p - p_{y_i},\ c = \mathbf{1} - e_{y_i},\ \mu = \frac{c+p}{\gamma + \|x_i\|^2/(\lambda n)},\ C = \frac{1}{1+\gamma\lambda n/\|x_i\|^2}$
  $a = \mathrm{OptimizeDual}(\mu, C)$
  $\alpha_i = -a, \alpha_{y_i} = \|a\|_1$
  $W = \hat{W} + \frac{1}{\lambda n} x_i \alpha_i^\top$
  Stopping condition:
   let $G = 0$
   for $i = 1, \ldots, n$
    $a = W^\top x_i, a = a - a_{y_i}, c = \mathbf{1} - e_{y_i}, b = \mathrm{Project}((a+c)/\gamma)$
    $G = G + \frac{\gamma}{2}(\|(a+c)/\gamma\|^2 - \|b - (a+c)/\gamma\|^2) + c^\top \alpha_i^{(t)} + \frac{\gamma}{2}(\|\alpha_i^{(t)}\|^2 - (\alpha_{i,y_i}^{(t)})^2)$
   Stop if $G/n + \lambda \mathrm{vec}(W)^\top \mathrm{vec}(W - Z) \leq \epsilon$

The runtime analysis is similar to SVM for the binary case.

6 Experiments

In this section we compare Prox-SDCA, its accelerated version Accelerated-Prox-SDCA, and the FISTA algorithm of [2], on $L_1 - L_2$ regularized loss minimization problems.

Following [26], we perform our experiments on three large datasets provided by Thorsten Joachims. For completeness, the characteristics of the datasets are given in Table 1.

These are binary classification problems, with each x_i being a vector which has been normalized to be $\|x_i\|_2 \leq 1$, and y_i being a binary class label of ± 1. We multiplied each x_i by y_i and following [26], we employed the smooth hinge loss, $\tilde{\phi}_\gamma$, as in (9), with $\gamma = 1$. The optimization problem we need to solve is therefore

$$\min_w P(w) \quad \text{where} \quad P(w) = \frac{1}{n}\sum_{i=1}^{n} \tilde{\phi}_\gamma(x_i^\top w) + \frac{\lambda}{2}\|w\|_2^2 + \sigma\|w\|_1.$$

In the experiments, we set $\sigma = 10^{-5}$ and vary λ in the range $\{10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}\}$.

The convergence behaviors are plotted in Fig. 3. In all the plots we depict the primal objective as a function of the number of passes over the data (often referred to as "epochs"). For FISTA, each iteration involves a single pass over the data. For Prox-SDCA, each n iterations are equivalent to a single pass over the data. And, for Accelerated-Prox-SDCA, each n inner iterations are equivalent to a single pass over the data. For Prox-SDCA and Accelerated-Prox-SDCA we implemented their corresponding stopping conditions and terminate the methods once an accuracy of 10^{-3} was guaranteed. Note that for Accelerated-Prox-SDCA, the stopping condition is often loose and convergence occurs much before the stopping condition is met.

It is clear from the graphs that Accelerated-Prox-SDCA yields the best results, and often significantly outperform the other methods. Prox-SDCA behaves similarly when λ is relatively large, but it converges much slower when λ is small. This is consistent with our theory, that shows that the convergence rate of Prox-SDCA depends on $1/\lambda$. Finally, the relative performance of FISTA and Prox-SDCA depends on the ratio between λ and n, but in all cases, Accelerated-Prox-SDCA is much faster than FISTA. This is again consistent with our theory.

Table 1 Summary of datasets

Dataset	Number of examples (n)	Number of features (d)	Sparsity ($\mathbb{E}_i[\|x_i\|_0/d]$) (%)
astro-ph	29,882	99,757	0.08
CCAT	781,265	47,236	0.16
cov1	522,911	54	22.22

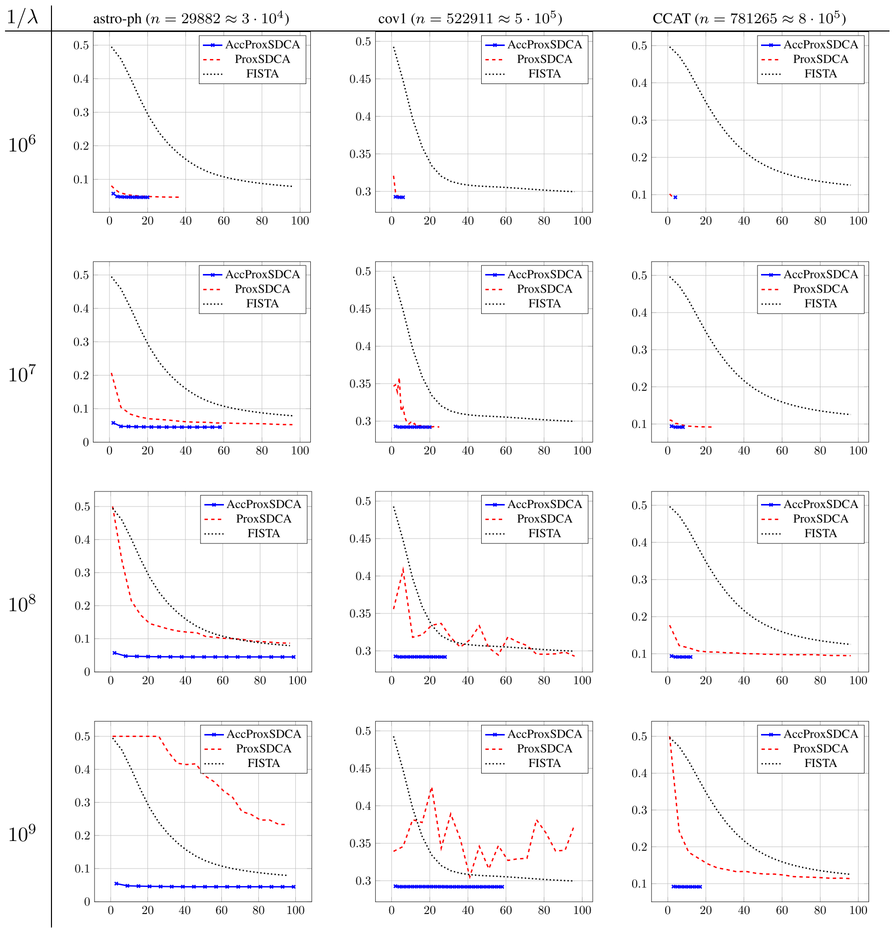

Fig. 3 Comparing Accelerated-Prox-SDCA, Prox-SDCA, and FISTA for minimizing the smoothed hinge-loss ($\gamma = 1$) with $L_1 - L_2$ regularization ($\sigma = 10^{-5}$ and λ varies in $\{10^{-6}, \ldots, 10^{-9}\}$). In each of these plots, the y-axis is the primal objective and the x-axis is the number of passes through the entire training set. The *three columns* corresponds to the three data sets. The methods are terminated either if stopping condition is met (with $\epsilon = 10^{-3}$) or after 100 passes over the data

7 Discussion and open problems

We have described and analyzed a proximal stochastic dual coordinate ascent method and have shown how to accelerate the procedure. The overall runtime of the resulting method improves state-of-the-art results in many cases of interest.

There are two main open problems that we leave to future research.

Open Problem 1 *When $\frac{1}{\lambda\gamma}$ is larger than n, the runtime of our procedure becomes $\tilde{O}\left(d\sqrt{\frac{n}{\lambda\gamma}}\right)$. Is it possible to derive a method whose runtime is $\tilde{O}\left(d\left(n + \sqrt{\frac{1}{\lambda\gamma}}\right)\right)$?*

Open Problem 2 *Our Prox-SDCA procedure and its analysis works for regularizers which are strongly convex with respect to an arbitrary norm. However, our acceleration procedure is designed for regularizers which are strongly convex with respect to the Euclidean norm. Is it possible to extend the acceleration procedure to more general regularizers?*

Acknowledgments The authors would like to thank Fen Xia for careful proof-reading of the paper which helped us to correct numerous typos. Shai Shalev-Shwartz is supported by the following grants: Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI) and ISF 598-10. Tong Zhang is supported by the following grants: NSF IIS-1407939 and NSF IIS-1250985.

Appendix: Proofs of iteration bounds for Prox-SDCA

The proof technique follows that of Shalev-Shwartz and Zhang [26], but with the required generality for handling general strongly convex regularizers and smoothness/Lipschitzness with respect to general norms. The proof of Theorem 1 is almost identical to the proof of Theorem 1 in Shalev-Shwartz and Zhang [25], except that we do not upper bound $\mathbb{E}[D(\alpha^*) - D(\alpha^{(0)})]$ by 1.

Proof of Theorem 2 Denote $\epsilon_D^{(t)} := D(\alpha^*) - D(\alpha^{(t)})$. Define $t_0 = \lceil \frac{n}{s} \log(2\epsilon_D^{(0)}/\epsilon_D)\rceil$. The proof of Theorem 1 implies that for every t, $\mathbb{E}[\epsilon_D^{(t)}] \le \epsilon_D^{(0)} e^{-\frac{st}{n}}$. By Markov's inequality, with probability of at least $1/2$ we have $\epsilon_D^{(t)} \le 2\epsilon_D^{(0)} e^{-\frac{st}{n}}$. Applying it for $t = t_0$ we get that $\epsilon_D^{(t_0)} \le \epsilon_D$ with probability of at least $1/2$. Now, lets apply the same argument again, this time with the initial dual sub-optimality being $\epsilon_D^{(t_0)}$. Since the dual is monotonically non-increasing, we have that $\epsilon_D^{(t_0)} \le \epsilon_D^{(0)}$. Therefore, the same argument tells us that with probability of at least $1/2$ we would have that $\epsilon_D^{(2t_0)} \le \epsilon_D$. Repeating this $\lceil \log_2(1/\delta)\rceil$ times, we obtain that with probability of at least $1 - \delta$, for some k we have that $\epsilon_D^{(kt_0)} \le \epsilon_D$. Since the dual is monotonically non-decreasing, the claim about the dual sub-optimality follows.

Next, for the duality gap, using the inequality

$$\mathbb{E}_t[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] \ge \frac{s}{n}\,(P(w^{(t-1)}) - D(\alpha^{(t-1)})).$$

from Lemma 1 of Shalev-Shwartz and Zhang [25].

We have that for every t such that $\epsilon_D^{(t-1)} \le \epsilon_D$ we have

$$P(w^{(t-1)}) - D(\alpha^{(t-1)}) \le \frac{n}{s}\,\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] \le \frac{n}{s}\epsilon_D.$$

This proves the second claim of Theorem 2.

For the last claim, suppose that at round T_0 we have $\epsilon_D^{(T_0)} \le \epsilon_D$. Let $T = T_0 + n/s$. It follows that if we choose t uniformly at random from $\{T_0, \ldots, T-1\}$, then $\mathbb{E}[P(w^{(t)}) - D(\alpha^{(t)})] \le \epsilon_D$. By Markov's inequality, with probability of at least $1/2$ we have $P(w^{(t)}) - D(\alpha^{(t)}) \le 2\epsilon_D$. Therefore, if we choose $\log_2(2/\delta)$ such random t, with probability $\ge 1 - \delta/2$, at least one of them will have $P(w^{(t)}) - D(\alpha^{(t)}) \le 2\epsilon_D$.

Combining with the first claim of the theorem, choosing $\epsilon_D = \epsilon_P/2$, and applying the union bound, we conclude the proof of the last claim of Theorem 2. $\square$

References

1. Baes, M.: Estimate Sequence Methods: Extensions and Approximations. Institute for Operations Research, ETH, Zürich (2009)
2. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
3. Collins, M., Globerson, A., Koo, T., Carreras, X., Bartlett, P.: Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. J. Mach. Learn. Res. **9**, 1775–1822 (2008)
4. Cotter, A., Shamir, O., Srebro, N., Sridharan, K.: Better mini-batch algorithms via accelerated gradient methods. arXiv preprint arXiv:1106.4574 (2011)
5. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. J. Mach. Learn. Res. **2**, 265–292 (2001)
6. d'Aspremont, A.: Smooth optimization with approximate gradient. SIAM J. Optim. **19**(3), 1171–1183 (2008)
7. Devolder, O., Glineur, F., Nesterov, Y.: First-order methods of smooth convex optimization with inexact oracle. Math. Program. **146**(1–2), 37–75 (2014)
8. Duchi, J., Singer, Y.: Efficient online and batch learning using forward backward splitting. J. Mach. Learn. Res. **10**, 2899–2934 (2009)
9. Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the l 1-ball for learning in high dimensions. In: Proceedings of the 25th International Conference on Machine Learning, pp. 272–279. ACM (2008)
10. Duchi, J., Shalev-Shwartz, S., Singer, Y., Tewari, A.: Composite objective mirror descent. In: Proceedings of the 23rd Annual Conference on Learning Theory, pp. 14–26 (2010)
11. Fercoq, O., Richtárik, P.: Accelerated, parallel and proximal coordinate descent. Technical report. arXiv:1312.5799 (2013)
12. Ghadimi, S., Lan, G.: Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework. SIAM J. Optim. **22**(4), 1469–1492 (2012)
13. Hu, C., Weike, P., Kwok, J.T.: Accelerated gradient methods for stochastic optimization and online learning. In: Advances in Neural Information Processing Systems, pp. 781–789 (2009)
14. Lacoste-Julien, S., Jaggi, M., Schmidt, M., Pletscher, P.: Stochastic block-coordinate frank-wolfe optimization for structural svms. arXiv preprint arXiv:1207.4747 (2012)
15. Langford, J., Li, L., Zhang, T.: Sparse online learning via truncated gradient. In: NIPS, pp. 905–912 (2009)
16. Roux, N.L., Schmidt, M., Bach, F.: A Stochastic Gradient Method with an Exponential Convergence Rate for Strongly-Convex Optimization with Finite Training Sets. arXiv preprint arXiv:1202.6258 (2012)
17. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J. Optim. **22**(2), 341–362 (2012)
18. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127–152 (2005)
19. Nesterov, Y.: Gradient methods for minimizing composite objective function. Math. Program. **140**, 125–161 (2013)
20. Richtárik, P., Takáč, M.: Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Math. Program. **144**(1–2), 1–38 (2014)
21. Schmidt, M., Roux, N.L., Bach, F.: Convergence rates of inexact proximal-gradient methods for convex optimization. Technical report. arXiv:1109.2415 (2011)
22. Shalev-Shwartz, S., Tewari, A.: Stochastic methods for l 1-regularized loss minimization. J. Mach. Learn. Res. **12**, 1865–1892 (2011)
23. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, Cambridge (2014)
24. Shalev-Shwartz, S., Tewari, A.: Stochastic methods for l 1-regularized loss minimization. In: ICML, p. 117 (2009)
25. Shalev-Shwartz, S., Zhang, T.: Proximal stochastic dual coordinate ascent. arXiv:1211.2717 (2012)

26. Shalev-Shwartz, S., Zhang, T.: Stochastic dual coordinate ascent methods for regularized loss minimization. J. Mach. Learn. Res. **14**, 567–599 (2013)
27. Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: primal estimated sub-GrAdient SOlver for SVM. In: ICML, pp. 807–814 (2007)
28. Shalev-Shwartz, S., Srebro, N., Zhang, T.: Trading accuracy for sparsity in optimization problems with sparsity constraints. SIAM J. Optim. **20**(6), 2807–2832 (2010)
29. Takác, M., Bijral, A., Richtárik, P., Srebro, N.: Mini-batch primal and dual methods for SVMs. In: ICML (2013)
30. Xiao, L.: Dual averaging method for regularized stochastic learning and online optimization. J. Mach. Learn. Res. **11**, 2543–2596 (2010)
31. Zhang, T.: On the dual formulation of regularized linear systems. Mach. Learn. **46**, 91–129 (2002)