Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization

Shai Shalev-Shwartz SHAIS@CS.HUJI.AC.IL
School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel

Tong Zhang TONGZ@RCI.RUTGERS.EDU
Department of Statistics, Rutgers University, NJ, USA, and Baidu Inc., Beijing, China

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 multi-class SVM. Experiments validate our theoretical findings.

1. Introduction

We consider the following generic optimization problem associated with regularized loss minimization of linear predictors: Let $X_1, \ldots, X_n$ be matrices in $\mathbb{R}^{d \times k}$ (referred to as instances), let $\phi_1, \ldots, \phi_n$ be a sequence of vector convex functions defined on $\mathbb{R}^k$ (referred to as loss functions), let $g(\cdot)$ be a convex function defined on $\mathbb{R}^d$ (referred to as a regularizer), and let $\lambda \geq 0$ (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

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

This applies, for example, to ridge regression and to logistic regression with L_2 regularization. The $\tilde{O}$ notation hides constants and logarithmic terms.

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 & Zhang [21], Le Roux et al. [13], 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 [21, 13]. It also significantly improves over the runtime of accelerated gradient descent due to Nesterov [16], 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. [23]), 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

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

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

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 Table 1 we specify the runtime of various algorithms (while ignoring constants and logarithmic terms) 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 $g(w) = \frac{1}{2}\|w\|_2^2$. Additional applications, and a more detailed runtime comparison to previous work, are given in Section 4. In the table, SGD stands for Stochastic Gradient Descent, and AGD stands for Accelerated Gradient Descent.

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 [21] 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 multiclass applications. As in [21], 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 & Zhang [21]. 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 & Takáč [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 Section 4. Recently, [11] 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. [6], Schmidt et al. [18], to allow approximate and stochastic proximal mapping. See also [1, 5]. In particular, it relies on similar ideas as in Proposition 4 of [18]. However, our specific requirement is different, and the proof presented here is different and significantly simpler than that of [18].

There have been several attempts to accelerate stochastic optimization algorithms. See for example [10, 9, 4] 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 [13, 21], 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 $f^*(y) = \sup_x \left[y^\top x - f(x)\right]$. 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

[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.

Problem	Algorithm	Runtime
SVM	SGD [23]	$\frac{d}{\lambda\epsilon}$
	AGD [15]	$dn\sqrt{\frac{1}{\lambda\epsilon}}$
	This paper	$d\left(n+\min\{\frac{1}{\lambda\epsilon},\sqrt{\frac{n}{\lambda\epsilon}}\}\right)$
Lasso	SGD and variants (e.g. [25, 24, 19])	$\frac{d}{\epsilon^2}$
	Stochastic Coordinate Descent [20, 14]	$\frac{dn}{\epsilon}$
	FISTA [16, 2]	$dn\sqrt{\frac{1}{\epsilon}}$
	This paper	$d\left(n+\min\{\frac{1}{\epsilon},\sqrt{\frac{n}{\epsilon}}\}\right)$
Ridge Regression	Exact	d^2n+d^3
	SGD [13], SDCA [21]	$d\left(n+\frac{1}{\lambda}\right)$
	AGD [16]	$dn\sqrt{\frac{1}{\lambda}}$
	This paper	$d\left(n+\min\{\frac{1}{\lambda},\sqrt{\frac{n}{\lambda}}\}\right)$

Table 1. The runtime of various algorithms for three key machine learning problems.

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 $f(w+v)\ge f(w)+\nabla f(w)^\top v+\frac{\gamma}{2}\|v\|_P^2$. 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 to maximize $D(\alpha)$ over $\alpha\in\mathbb{R}^{k\times n}$ where

$$D(\alpha)=\left[\frac{1}{n}\sum_{i=1}^n -\phi_i^*(-\alpha_i)-\lambda g^*\left(\frac{1}{\lambda n}\sum_{i=1}^n X_i\alpha_i\right)\right], \tag{2}$$

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. If we define

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

then it is known that $w(\alpha^*)=w^*$, where α^* is an optimal solution of (2). It is also known that $P(w^*)=D(\alpha^*)$ which immediately implies that for all w and α, we have $P(w)\ge D(\alpha)$, and hence the **duality gap** defined as $P(w(\alpha))-D(\alpha)$ can be regarded as an upper bound on both the **primal sub-optimality**, $P(w(\alpha))-P(w^*)$, and on the **dual sub-optimality**, $D(\alpha^*)-D(\alpha)$.

3. Main Results

In this section we describe our algorithms and their analysis. We start in Section 3.1 with a description of our proximal stochastic dual coordinate ascent procedure (Prox-SDCA). Then, in Section 3.2 we show how to accelerate the method by calling Prox-SDCA on a sequence of problems with a strong regularization. Throughout this section we assume that the loss functions are smooth. The case of non-smooth but Lipschitz loss functions can be tackled by applying a "smoothing" technique (see Nesterov [15]).

Due to the lack of space, all proofs are omitted from this extended abstract and can be found in the long version of the paper [22]. The long version also contains detailed pseudo-code of all the algorithms.

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 for g being 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

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

Note that this particular update is rather similar to the update step of proximal-gradient dual-averaging method (see for example Xiao [24]). 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 $\Delta\alpha_i$ be the maximizer of

$$-\frac{1}{n}\phi_i^*(-(\alpha_i + \Delta\alpha_i)) - \lambda g^*(v^{(t-1)} + (\lambda n)^{-1} X_i \Delta\alpha_i).$$

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 a proximal objective which is a lower bound of the dual objective. This yields maximization of the expression:

$$-\phi_i^*(-(\alpha_i + \Delta\alpha_i)) - w^{(t-1)\top} X_i \Delta\alpha_i - \frac{1}{2\lambda n} \|X_i \Delta\alpha_i\|_{D'}^2 .$$

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 setting $s = \frac{\lambda n \gamma}{R^2 + \lambda n \gamma}$, for R being an upper bound on $\|X_i\|_{D \to D'}$, still leads to a sufficient increase in the dual objective. A detailed pseudo-code can be found in [22].

The theorem below provides an upper bound on the number of iterations required by our prox-SDCA procedure.

Theorem 1. *The expected runtime required to minimize P up to accuracy ϵ using procedure Prox-SDCA is*

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

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 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" $\beta(w^{(t-1)} - w^{(t-2)})$.

The values of β and κ are set by our theoretical analysis as follows: $\kappa = R^2/(\gamma n) - \lambda$, and $\beta = \frac{1-\eta}{1+\eta}$ where $\eta^{-1} = \sqrt{-1 + \kappa/\lambda}$. At each "outer" iteration of the acceleration procedure, we apply Prox-SDCA for approximately solving $\tilde{P}_t(w)$. We initialize the dual solution to be the dual solution from the previous iteration, and we require the accuracy of Prox-SDCA at iteration t to be $\frac{\eta}{2(1+\eta^{-1})}\xi_{t-1}$ where $\xi_1 = (1 + \eta^{-2})(P(0) - D(0))$ and $\xi_t = (1 - \eta/2)^{t-1}\xi_{t-1}$.

A detailed pseudo-code of the algorithm is given in [22]. All the parameters of the algorithm are determined by our theory.

Theorem 2. *The total runtime required by accelerated Prox-SDCA to guarantee an ϵ-accurate solution with probability of at least $1 - \delta$ is*

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

4. Applications

In this section we specify our algorithmic framework to several popular machine learning applications. In Section 4.1 we start by describing several loss functions and

deriving their conjugate. In Section 4.2 we describe several regularization functions. Finally, in the rest of the subsections we specify our algorithm for Ridge regression, SVM, and Lasso.

4.1. Loss functions

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

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

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. This loss is parameterized by a scalar $\gamma > 0$ and is defined as:

$$\tilde{\phi}_\gamma(a) = \begin{cases} 0 & a \geq 1 \\ 1 - a - \gamma/2 & a \leq 1 - \gamma \\ \frac{1}{2\gamma}(1-a)^2 & \text{o.w.} \end{cases} \tag{4}$$

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

4.2. 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

$$g(w) = \frac{1}{2}\|w\|_2^2 + \sigma' \|w\|_1 , \tag{5}$$

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

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

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 $\nabla_i g^*(v) = \operatorname{sign}(v_i)\left[|v_i| - \sigma'\right]_+$. It follows that $g^*(v) = \frac{1}{2}\sum_i \left(\left[|v_i| - \sigma'\right]_+\right)^2$.

4.3. 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

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

The runtime of Prox-SDCA for ridge regression is

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

where $R = \max_i \|x_i\|$. This matches the recent results of [13, 21]. If $R^2/\lambda \gg n$ we can apply the accelerated procedure and obtain the improved runtime

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

4.4. 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:

$$\min_w \left[\frac{1}{2n}\sum_{i=1}^n (x_i^\top w - y_i)^2 + \sigma\|w\|_1\right] , \qquad (6)$$

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

Consider the optimization problem of minimizing

$$P(w) = \frac{1}{2n}\sum_{i=1}^n (x_i^\top w - y_i)^2 + \lambda\left(\frac{1}{2}\|w\|_2^2 + \frac{\sigma}{\lambda}\|w\|_1\right) , \qquad (7)$$

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 (7), then it is easy to verify that setting $\lambda = \epsilon(\sigma/\bar{y})^2$ guarantees that w^* is an ϵ accurate solution to the original problem given in (6).

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

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

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

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

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. [12, 19, 24, 7, 8]). 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 (6) is stochastic coordinate descent over the primal problem. [19] 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 [14].

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 much better than that of [19]. 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 the same or better than [19] whenever $d = O(n)$.

4.5. 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 (4). Recall that the smooth hinge-loss satisfies, for every 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.

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

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

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

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

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

5. 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.

The experiments were performed on three large datasets with very different feature counts and sparsity, which were kindly provided by Thorsten Joachims (the datasets were also used in [21]). These are binary classification problems, with each x_i being a vector which has been normalized to be $\|x_i\|_2 = 1$, and y_i being a binary class label of ± 1. We multiplied each x_i by y_i and following [21], we employed the smooth hinge loss, $\tilde{\phi}_\gamma$, as in (4), with $\gamma = 1$. The optimization problem we need to solve is therefore to minimize

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

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 Figure 1. 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.

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. 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.

6. 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 is possible to extend the acceleration procedure to more general regularizers?*

Acknowledgements

The authors would like to thank Fen Xia for careful proof-reading of the paper which helped us 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-1016061, NSF DMS-1007527, and NSF IIS-1250985.

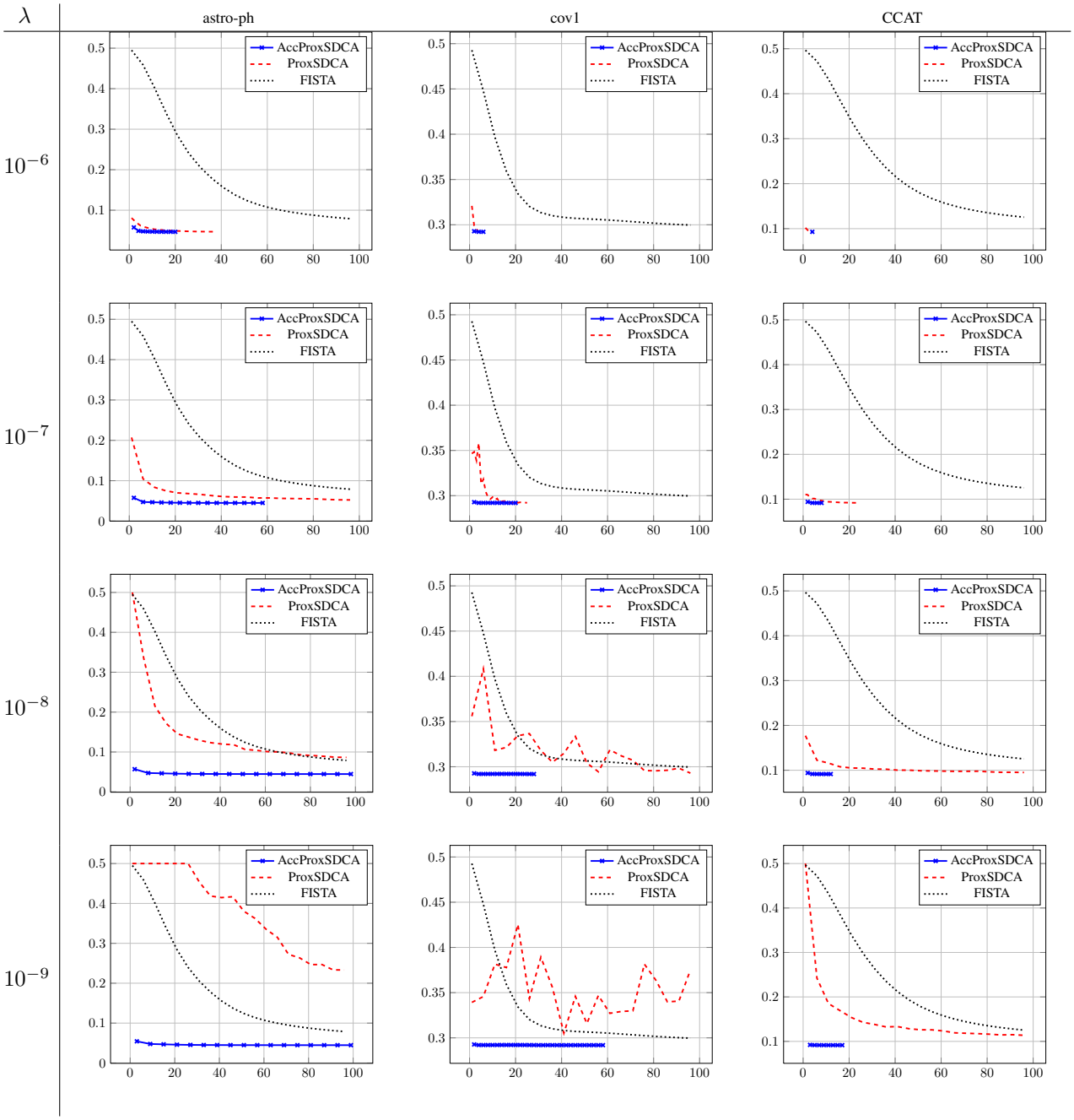

Figure 1. Comparing Accelerated-Prox-SDCA, Prox-SDCA, and FISTA for minimizing the smoothed hinge-loss ($\gamma = 1$) with $L_1 - L_2$ regularization ($\sigma = 10^{-5}$ and λ varies in $\{10^{-6}, \ldots, 10^{-9}\}$). In each of these plots, the y-axis is the primal objective and the x-axis is the number of passes through the entire training set. The three columns corresponds to the three data sets described in [21]. The methods are terminated either if stopping condition is met (with $\epsilon = 10^{-3}$) or after 100 passes over the data.

References

[1] Baes, Michel. Estimate sequence methods: extensions and approximations. *Institute for Operations Research, ETH, Zürich, Switzerland*, 2009.

[2] Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[3] Collins, M., A. Globerson, Koo, T., Carreras, X., and Bartlett, P. Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. *Journal of Machine Learning Research*, 9: 1775–1822, 2008.

[4] Cotter, Andrew, Shamir, Ohad, Srebro, Nathan, and Sridharan, Karthik. Better mini-batch algorithms via accelerated gradient methods. *arXiv preprint arXiv:1106.4574*, 2011.

[5] d'Aspremont, Alexandre. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008.

[6] Devolder, Olivier, Glineur, Francois, and Nesterov, Yuri. First-order methods of smooth convex optimization with inexact oracle. Technical Report 2011/2, CORE, 2011.

[7] Duchi, J. and Singer, Y. Efficient online and batch learning using forward backward splitting. *The Journal of Machine Learning Research*, 10:2899–2934, 2009.

[8] Duchi, John, Shalev-Shwartz, Shai, Singer, Yoram, and Tewari, Ambuj. Composite objective mirror descent. In *Proceedings of the 23rd Annual Conference on Learning Theory*, pp. 14–26, 2010.

[9] Ghadimi, Saeed and Lan, Guanghui. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

[10] Hu, Chonghai, Pan, Weike, and Kwok, James T. Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems*, pp. 781–789, 2009.

[11] Lacoste-Julien, S., Jaggi, M., Schmidt, M., and Pletscher, P. Stochastic block-coordinate frank-wolfe optimization for structural svms. *arXiv preprint arXiv:1207.4747*, 2012.

[12] Langford, J., Li, L., and Zhang, T. Sparse online learning via truncated gradient. In *NIPS*, pp. 905–912, 2009.

[13] Le Roux, Nicolas, Schmidt, Mark, and Bach, Francis. A Stochastic Gradient Method with an Exponential Convergence Rate for Strongly-Convex Optimization with Finite Training Sets. *arXiv preprint arXiv:1202.6258*, 2012.

[14] Nesterov, Y. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[15] Nesterov, Yurii. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[16] Nesterov, Yurii. Gradient methods for minimizing composite objective function, 2007.

[17] Richtárik, Peter and Takáč, Martin. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, pp. 1–38, 2012.

[18] Schmidt, Mark, Roux, Nicolas Le, and Bach, Francis. Convergence rates of inexact proximal-gradient methods for convex optimization. Technical Report arXiv:1109.2415, arXiv, 2011.

[19] Shalev-Shwartz, S. and Tewari, A. Stochastic methods for l 1-regularized loss minimization. *The Journal of Machine Learning Research*, 12:1865–1892, 2011.

[20] Shalev-Shwartz, Shai and Tewari, Ambuj. Stochastic methods for l_1 regularized loss minimization. In *ICML*, pp. 117, 2009.

[21] Shalev-Shwartz, Shai and Zhang, Tong. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, Feb 2013.

[22] Shalev-Shwartz, Shai and Zhang, Tong. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. arxiv:1309.2375, 2013.

[23] Shalev-Shwartz, Shai, Singer, Yoram, and Srebro, Nathan. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *ICML*, pp. 807–814, 2007.

[24] Xiao, Lin. Dual averaging method for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.

[25] Zhang, Tong. On the dual formulation of regularized linear systems. *Machine Learning*, 46:91–129, 2002.