Abstract
This paper introduces AdaSDCA: an adaptive variant of stochastic dual coordinate ascent (SDCA) for solving the regularized empirical risk minimization problems. Our modification consists in allowing the method to adaptively change the probability distribution over the dual variables throughout the iterative process. AdaSDCA achieves provably better complexity bound than SDCA with the best fixed probability distribution, known as importance sampling. However, it is of a theoretical character as it is expensive to implement. We also propose AdaSDCA+: a practical variant which in our experiments outperforms existing non-adaptive methods.

1. Introduction

Empirical Loss Minimization. In this paper we consider the regularized empirical risk minimization problem:

$$
\min_{w \in \mathbb{R}^d} \left[ P(w) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \phi_i(A_i^T w) + \lambda g(w) \right].
$$

(1)

In the context of supervised learning, $w$ is a linear predictor, $A_1, \ldots, A_n \in \mathbb{R}^d$ are samples, $\phi_1, \ldots, \phi_n : \mathbb{R} \to \mathbb{R}$ are loss functions, $g : \mathbb{R}^d \to \mathbb{R}$ is a regularizer and $\lambda > 0$ a regularization parameter. Hence, we are seeking to identify the predictor which minimizes the average (empirical) loss $P(w)$.

We assume throughout that the loss functions are $1/\gamma$-smooth for some $\gamma > 0$. That is, we assume they are differentiable and have Lipschitz derivative with Lipschitz constant $1/\gamma$:

$$
|\phi'(a) - \phi'(b)| \leq \frac{1}{\gamma} |a - b|
$$

for all $a, b \in \mathbb{R}$. Moreover, we assume that $g$ is 1-strongly convex with respect to the L2 norm:

$$
g(w) \leq \alpha g(w_1) + (1 - \alpha) g(w_2) - \frac{\alpha(1 - \alpha)}{2} \|w_1 - w_2\|^2
$$

for all $w_1, w_2 \in \text{dom } g$, $0 \leq \alpha \leq 1$ and $w = \alpha w_1 + (1 - \alpha) w_2$.

The ERM problem (1) has received considerable attention in recent years due to its widespread usage in supervised statistical learning (Shalev-Shwartz & Zhang, 2013b). Often, the number of samples $n$ is very large and it is important to design algorithms that would be efficient in this regime.

Modern stochastic algorithms for ERM. Several highly efficient methods for solving the ERM problem were proposed and analyzed recently. These include primal methods such as SAG (Schmidt et al., 2013), SVRG (Johnson & Zhang, 2013), S2GD (Konečný & Richtárik, 2014), SAGA (Defazio et al., 2014), mS2GD (Konečný et al., 2014a) and MISO (Mairal, 2015). Importance sampling was considered in ProxSVRG (Xiao & Zhang, 2014) and S2CD (Konečný et al., 2014b).

Stochastic Dual Coordinate Ascent. One of the most successful methods in this category is stochastic dual coordinate ascent (SDCA), which operates on the dual of the ERM problem (1):
where functions $f$ and $\psi$ are defined by

$$f(\alpha) \triangleq \lambda g^* \left( \frac{1}{\lambda n} \sum_{i=1}^{n} A_i \alpha_i \right), \quad (3)$$

$$\psi(\alpha) = \frac{1}{n} \sum_{i=1}^{n} \phi_i^* (\alpha_i), \quad (4)$$

and $g^*$ and $\phi_i^*$ are the convex conjugates\(^1\) of $g$ and $\phi_i$, respectively. Note that in dual problem, there are as many variables as there are samples in the primal: $\alpha \in \mathbb{R}^n$.

SDCA in each iteration randomly selects a dual variable $\alpha_i$, and performs its update, usually via closed-form formula – this strategy is know as randomized coordinate descent. Methods based on updating randomly selected dual variables enjoy, in our setting, a linear convergence rate. Methods based on updating randomly selected dual variables enjoy, in our setting, a linear convergence rate. Methods based on updating randomly selected dual variables enjoy, in our setting, a linear convergence rate. Methods based on updating randomly selected dual variables enjoy, in our setting, a linear convergence rate.

We now briefly highlight the main contributions of this work.

Two algorithms with adaptive probabilities. We propose two new stochastic dual ascent algorithms: AdaSDCA (Algorithm 1) and AdaSDCA+ (Algorithm 2) for solving (1) and its dual problem (2). The novelty of our algorithms is in adaptive choice of the probability distribution over individual coordinates and even subsets of coordinates (Richářík & Takáč, 2013b; Qu et al., 2014; Qu & Richářík, 2014; Qu & Richářík, 2014). In all of these works the theory allows the computation of a fixed probability distribution, known as importance sampling, which optimizes the complexity bounds. However, such a distribution often depends on unknown quantities, such as the distances of the individual variables from their optimal values (Richářík & Takáč, 2014; Qu & Richářík, 2014). In some cases, such as for smooth strongly convex functions or in the primal-dual setup we consider here, the probabilities forming an importance sampling can be explicitly computed (Richářík & Takáč, 2013b; Zhao & Zhang, 2015; Qu et al., 2014; Qu & Richářík, 2014; Qu & Richářík, 2014). Typically, the theoretical influence of using the importance sampling is in the replacement of the maximum of certain data-dependent quantities in the complexity bound by the average.

Adaptivity. Despite the striking developments in the field, there is virtually no literature on methods using an adaptive choice of the probabilities. We are aware of a few pieces of work; but all resort to heuristics unsupported by theory (Glasmachers & Dogan, 2013; Lukasewitz, 2013; Schaul et al., 2013; Banks-Watson, 2012; Loshchilov et al., 2014; Tappenden et al., 2014), which unfortunately also means that the methods are sometimes effective, and sometimes not. We observe that in the primal-dual framework we consider, each dual variable can be equipped with a natural measure of progress which we call “dual residue”. We propose that the selection probabilities be constructed based on these quantities.

Outline: In Section 2 we summarize the contributions of our work. In Section 3 we describe our first, theoretical methods (Algorithm 1: AdaSDCA) and describe the intuition behind it. In Section 4 we provide convergence analysis. In Section 5 we introduce AdaSDCA+ (Algorithm 2): a variant of AdaSDCA (Algorithm 1) containing heuristic elements which make it efficiently implementable. We conclude with numerical experiments in Section 6. Technical proofs and additional numerical experiments can be found in the supplementary materials.

We observe that in the primal-dual framework we consider, each dual variable can be equipped with a natural measure of progress which we call “dual residue”. We propose that the selection probabilities be constructed based on these quantities.

2. Contributions

\(^1\)By the convex (Fenchel) conjugate of a function $h : \mathbb{R}^k \rightarrow \mathbb{R}$ we mean the function $h^* : \mathbb{R}^k \rightarrow \mathbb{R}$ defined by $h^*(u) = \sup_s \{ s^T u - h(s) \}$. 

Stochastic Dual Coordinate Ascent with Adaptive Probabilities
Practical method. AdaSDCA requires the same computational effort per iteration as the batch gradient algorithm. To solve this issue, we propose AdaSDCA+ (Algorithm 2): an efficient heuristic variant of the AdaSDCA. The computational effort of the heuristic method in a single iteration is low, which makes it very competitive with methods based on importance sampling, such as IProx-SDCA (Zhao & Zhang, 2015). We support this with computational experiments in Section 6.

3. The Algorithm: AdaSDCA

It is well known that the optimal primal-dual pair \((w^*, \alpha^*) \in \mathbb{R}^d \times \mathbb{R}^n\) satisfies the following optimality conditions:

\[
w^* = \nabla g^* \left( \frac{1}{\lambda} A\alpha^* \right) \tag{5}
\]

\[\alpha_i^* = -\nabla \phi_i(A_i^T w^*), \quad \forall i \in [n] \overset{\text{def}}{=} \{1, \ldots, n\}, \tag{6}\]

where \(A\) is the \(d\)-by-\(n\) matrix with columns \(A_1, \ldots, A_n\).

Definition 1 (Dual residue). The dual residue, \(\kappa = (\kappa_1, \ldots, \kappa_n) \in \mathbb{R}^n\), associated with \((w, \alpha)\) is given by:

\[\kappa_i \overset{\text{def}}{=} \alpha_i + \nabla \phi_i(A_i^T w). \tag{7}\]

Note, that \(\kappa_i^2 = 0\) if and only if \(\alpha_i\) satisfies (5). This motivates the design of AdaSDCA (Algorithm 1) as follows: whenever \(|\kappa_i|\) is large, the \(i\)th dual coordinate \(\alpha_i\) is suboptimal and hence should be updated more often.

Definition 2 (Coherence). We say that probability vector \(p^t \in \mathbb{R}^n\) is coherent with the dual residue \(\kappa^t\) if for all \(i \in [n]\) we have

\[\kappa_i^t \neq 0 \quad \Rightarrow \quad p_i^t > 0. \tag{9}\]

Alternatively, \(p^t\) is coherent with \(\kappa^t\) if for

\[I_t \overset{\text{def}}{=} \{i \in [n]: \kappa_i^t \neq 0\} \subseteq [n]. \tag{10}\]

we have \(\min_{i \in I_t} p_i^t > 0\).

AdaSDCA is a stochastic dual coordinate ascent method, with an adaptive probability vector \(p^t\), which could potentially change at every iteration \(t\). The primal and dual update rules are exactly the same as in standard SDCA (Shalev-Shwartz & Zhang, 2013b), which instead uses uniform sampling probability at every iteration and does not require the computation of the dual residue \(\kappa\).

Our first result highlights a key technical tool which ultimately leads to the development of good adaptive sampling distributions \(p^t\) in AdaSDCA. For simplicity we denote by \(E_t\) the expectation with respect to the random index \(i_t \in [n]\) generated at iteration \(t\).

Algorithm 1 AdaSDCA

\textbf{Init:} \(v_i = A_i^T A_i\) for \(i \in [n]; \alpha^0 \in \mathbb{R}^n; \bar{\alpha}^0 = \frac{1}{\lambda n} A\alpha^0\)

\textbf{for} \(t \geq 0\) \textbf{do}

\textbf{Primal update:} \(w^t = \nabla g^* (\bar{\alpha}^t)\)

\textbf{Set:} \(\alpha^{t+1} = \alpha^t\)

\textbf{Compute residue:} \(\kappa^t_i = \alpha^t_i + \nabla \phi_i(A_i^T w^t), \forall i \in [n]\)

\textbf{Compute probability distribution:} \(p^t\) coherent with \(\kappa^t\)

\textbf{Generate random} \(i_t \in [n]\) according to \(p^t\)

\textbf{Compute:}

\[\Delta \alpha_i^t = \arg \max_{\Delta \in \mathbb{R}} \left\{ -\phi_i^* \left( -\left( \alpha_i^t + \Delta \right) \right) - A_i^T w^t \Delta - \frac{\nu_i}{2\lambda n} |\Delta|^2 \right\} \]

\textbf{Dual update:} \(\alpha_i^{t+1} = \alpha_i^t + \Delta \alpha_i^t\)

\textbf{Average update:} \(\bar{\alpha}^t = \bar{\alpha}^t + \frac{\Delta \alpha_i^t}{\lambda n} A_i^T\)

\textbf{end for}

\textbf{Output:} \(w^t, \alpha^t\)

\textbf{Lemma 3.} Consider the AdaSDCA algorithm during iteration \(t \geq 0\) and assume that \(p^t\) is coherent with \(\kappa^t\). Then

\[E_t \left[ D(\alpha^{t+1}) - D(\alpha^t) \right] - \theta \left( P(w^t) - D(\alpha^t) \right) \geq -\frac{\theta}{2\lambda n^2} \sum_{i \in I_t} \left( \frac{\theta (v_i + n\lambda \gamma)}{p_i^t} - n\lambda \gamma \right) |\kappa_i^t|^2, \tag{8}\]

for arbitrary

\[0 \leq \theta \leq \min_{i \in I_t} p_i^t. \tag{9}\]

\textbf{Proof.} Lemma 3 is proved similarly to Lemma 2 in (Zhao & Zhang, 2015), but in a slightly more general setting. For completeness, we provide the proof in the supplementary materials.

\[\theta(\kappa, p) \equiv \frac{n \lambda \gamma \sum_{i: i \in I_t} |\kappa_i^t|^2}{\sum_{i: i \in I_t} p_i^t |\kappa_i^t|^2 (v_i + n\lambda \gamma)}. \tag{10}\]

We also need to make sure that \(0 \leq \theta \leq \min_{i \in I_t} p_i^t\) in order to apply Lemma 3. A “good” adaptive probability \(p^t\) should then be the solution of the following optimization problem:
\[
\max_{p \in \mathbb{R}^n_+} \theta(\kappa^t, p) \quad (11)
\]
\[
\text{s.t. } \sum_{i=1}^{n} p_i = 1 \quad \theta(\kappa^t, p) \leq \min_{i: \kappa^t_i \neq 0} p_i
\]
A feasible solution to (11) is the importance sampling (also known as optimal serial sampling) \(\bar{p}\) defined by:
\[
\bar{p}_i \defeq \frac{v_i + n\lambda^\gamma}{\sum_{j=1}^{n} (v_j + n\lambda^\gamma)}, \quad \forall i \in [n], \quad (12)
\]
which was proposed in (Zhao & Zhang, 2015) to obtain proximal stochastic dual coordinate ascent method with importance sampling (IProx-SDCA). The same optimal probability vector was also deduced, via different means and in a more general setting in (Qu et al., 2014). Note that in this special case, since \(p^*\) is independent of the residue \(\kappa^t\), the computation of \(\kappa^t\) is unnecessary and hence the complexity of each iteration does not scale up with \(n\).

It seems difficult to identify other feasible solutions to program (11) apart from \(p^*\), not to mention solve it exactly. However, by relaxing the constraint \(\theta(\kappa^t, p^*) \leq \min_{i: \kappa^t_i \neq 0} p_i\), we obtain an explicit optimal solution.

**Lemma 4.** The optimal solution \(p^*(\kappa^t)\) of
\[
\max_{p \in \mathbb{R}^n_+} \theta(\kappa^t, p)
\]
\[
\text{s.t. } \sum_{i=1}^{n} p_i = 1
\]
is:
\[
(p^*(\kappa^t))_i = \frac{|\kappa^t_i| \sqrt{v_i + n\lambda^\gamma}}{\sum_{j=1}^{n} |\kappa^t_j| \sqrt{v_j + n\lambda^\gamma}}, \quad \forall i \in [n]. \quad (14)
\]

**Proof.** The proof is deferred to the supplementary materials.

The suggestion made by (14) is clear: we should update more often those dual coordinates \(\alpha_i\) which have large absolute dual residue \(|\kappa^t_i|\) and/or large Lipschitz constant \(v_i\).

If we let \(p^* = p^*(\kappa^t)\) and \(\theta = \theta(\kappa^t, p^*)\), the constraint (9) may not be satisfied, in which case (8) does not necessarily hold. However, as shown by the next lemma, the constraint (9) is not required for obtaining (8) when all the functions \(\phi_i\) are quadratic.

**Lemma 5.** Suppose that all \(\{\phi_i\}\) are quadratic. Let \(t \geq 0\).
If \(\min_{i \in I^t} p^t_i > 0\), then (8) holds for any \(\theta \in [0, +\infty)\).

The proof is deferred to the supplementary materials.

### 4. Convergence results

In this section we present our theoretical complexity results for AdaSDCA. The main results are formulated in Theorem 7, covering the general case, and in Theorem 11 in the special case when \(\{\phi_i\}\) are all quadratic.

#### 4.1. General loss functions

We derive the convergence result from Lemma 3.

**Proposition 6.** Let \(t \geq 0\). If \(\min_{i \in I^t} p^t_i > 0\) and \(\theta(\kappa^t, p^t) \leq \min_{i \in I^t} p^t_i\), then
\[
\mathbb{E}_t [D(\alpha^{t+1}) - D(\alpha^t)] \geq \theta(\kappa^t, p^t) \left( P(w^t) - D(\alpha^t) \right).
\]

**Proof.** This follows directly from Lemma 3 and the fact that the right-hand side of (8) equals 0 when \(\theta = \theta(\kappa^t, p^t)\).

**Theorem 7.** Consider AdaSDCA. If at each iteration \(t \geq 0\), \(\min_{i \in I^t} p^t_i > 0\) and \(\theta(\kappa^t, p^t) \leq \min_{i \in I^t} p^t_i\), then
\[
\mathbb{E}[P(w^t) - D(\alpha^t)] \leq \frac{1}{\theta_t} \prod_{k=0}^{t} (1 - \tilde{\theta}_k) \left( D(\alpha^t) - D(\alpha^0) \right),
\]
for all \(t \geq 0\) where
\[
\tilde{\theta}_t \defeq \frac{\mathbb{E}[\theta(\kappa^t, p^t)(P(w^t) - D(\alpha^t))]}{\mathbb{E}[P(w^t) - D(\alpha^t)]}. \quad (16)
\]

**Proof.** By Proposition 6, we know that
\[
\mathbb{E}[D(\alpha^{t+1}) - D(\alpha^t)] \geq \mathbb{E}[\theta(\kappa^t, p^t)(P(w^t) - D(\alpha^t))]
\]
\[
\overset{(16)}{=} \tilde{\theta}_t \mathbb{E}[P(w^t) - D(\alpha^t)] \geq \tilde{\theta}_t \mathbb{E}[D(\alpha^t) - D(\alpha^0)],
\]
whence
\[
\mathbb{E}[D(\alpha^t) - D(\alpha^{t+1})] \leq (1 - \tilde{\theta}_t) \mathbb{E}[D(\alpha^t) - D(\alpha^0)].
\]
Therefore,
\[
\mathbb{E}[D(\alpha^t) - D(\alpha^0)] \leq \prod_{k=0}^{t} (1 - \tilde{\theta}_k) \left( D(\alpha^t) - D(\alpha^0) \right).
\]
By plugging the last bound into (17) we get the bound on the primal dual error:
\[
\mathbb{E}[P(w^t) - D(\alpha^t)] \leq \frac{1}{\theta_t} \mathbb{E}[D(\alpha^{t+1}) - D(\alpha^t)]
\]
\[
\leq \frac{1}{\theta_t} \mathbb{E}[D(\alpha^t) - D(\alpha^0)]
\]
\[
\leq \frac{1}{\theta_t} \prod_{k=0}^{t} (1 - \tilde{\theta}_k) \left( D(\alpha^t) - D(\alpha^0) \right). \quad \square
\]
As mentioned in Section 3, by letting every sampling probability \( p^t \) be the importance sampling (optimal serial sampling) \( p^* \) defined in (12), AdaSDCA reduces to IProx-SDCA proposed in (Zhao & Zhang, 2015). The convergence theory established for IProx-SDCA in (Zhao & Zhang, 2015), which can also be derived as a direct corollary of our Theorem 7, is stated as follows.

**Theorem 8 (Zhao & Zhang, 2015).** Consider AdaSDCA with \( p^t = \tilde{p} \) defined in (12) for all \( t \geq 0 \). Then

\[
\mathbb{E}[P(w^t) - D(\alpha^t)] \geq \frac{1}{\tilde{\theta}} (1 - \tilde{\theta})^t (D(\alpha^*) - D(\alpha^0)),
\]

where

\[
\tilde{\theta} = \frac{n\lambda\gamma}{\sum_{i=1}^n (\nu_i + \lambda\gamma n)}.
\]

The next corollary suggests that a better convergence rate than IProx-SDCA can be achieved by using properly chosen adaptive sampling probability.

**Corollary 9.** Consider AdaSDCA. If at each iteration \( t \geq 0 \), \( p^t \) is the optimal solution of (11), then (15) holds and \( \tilde{\theta}_t \geq \theta \) for all \( t \geq 0 \).

However, solving (11) requires large computational effort, because of the dimension \( n \) and the non-convex structure of the program. We show in the next section that when all the loss functions \( \{\phi_i\}_i \) are quadratic, then we can get better convergence rate in theory than IProx-SDCA by using the optimal solution of (13).

### 4.2. Quadratic loss functions

The main difficulty of solving (11) comes from the inequality constraint, which originates from (9). In this section we mainly show that the constraint (9) can be released if all \( \{\phi_i\}_i \) are quadratic.

**Proposition 10.** Suppose that all \( \{\phi_i\}_i \) are quadratic. Let \( t \geq 0 \). If \( \min_{i \in T} p^t_i > 0 \), then

\[
\mathbb{E}_t[D(\alpha^{t+1}) - D(\alpha^t)] \geq \theta(\kappa^t, p^t) (P(w^t) - D(\alpha^t)).
\]

**Proof.** This is a direct consequence of Lemma 5 and the fact that the right-hand side of (8) equals 0 when \( \theta = \theta(\kappa^t, p^t) \).

**Theorem 11.** Suppose that all \( \{\phi_i\}_i \) are quadratic. Consider AdaSDCA. If at each iteration \( t \geq 0 \), \( \min_{i \in T} p^t_i > 0 \), then (15) holds for all \( t \geq 0 \).

**Proof.** We only need to apply Proposition 10. The rest of the proof is the same as in Theorem 7.

**Corollary 12.** Suppose that all \( \{\phi_i\}_i \) are quadratic. Consider AdaSDCA. If at each iteration \( t \geq 0 \), \( p^t \) is the optimal solution of (13), which has a closed form (14), then (15) holds for

\[
\tilde{\theta}_t = \frac{\mathbb{E} \left[ \frac{n\lambda\gamma \sum_{i=1}^n |v_i|^2}{\sum_{i=1}^n |v_i|^2 + (n+\lambda\gamma)^2} (P(w^t) - D(\alpha^t)) \right]}{\mathbb{E}[P(w^t) - D(\alpha^t)]} \geq \tilde{\theta}.
\]

**Remark 13.** The latter inequality follows easily from Cauchy-Schwarz inequality. The gap between \( \tilde{\theta}_t \) of our method and \( \theta \) of IProx-SDCA depends on the proportion of the dual residue \( \kappa^t \). We leave a quantitative comparison between \( \tilde{\theta}_t \) and \( \theta \) for future work.

### 5. Efficient heuristic variant

Corollary 9 and 12 suggest how to choose adaptive sampling probability in AdaSDCA which yields a theoretical convergence rate at least as good as IProx-SDCA (Zhao & Zhang, 2015). However, there are two main implementation issues of AdaSDCA:

1. The update of the dual residue \( \kappa^t \) at each iteration costs \( O(\text{nnz}(A)) \) where \( \text{nnz}(A) \) is the number of nonzero elements of the matrix \( A \);
2. We do not know how to compute the optimal solution of (11).

In this section, we propose a heuristic variant of AdaSDCA, which avoids the above two issues while staying close to the ‘good’ adaptive sampling distribution.

### 5.1. Description of Algorithm

AdaSDCA+ has the same structure as AdaSDCA with a few important differences.

**Epochs** AdaSDCA+ is divided into epochs of length \( n \). At the beginning of every epoch, sampling probabilities are computed according to one of two options. During each epoch the probabilities are cheaply updated at the end of every iteration to approximate the adaptive model. The intuition behind is as follows. After \( i \) is sampled and the dual coordinate \( \alpha_i \) is updated, the residue \( \kappa_i \) naturally decreases. We then decrease also the probability that \( i \) is chosen in the next iteration, by setting \( p_i^{t+1} \) to be proportional to \( (p_i^t, \ldots, p_{i-1}^t, p_i^t/m, p_{i+1}^t, \ldots, p_n^t) \). By doing this we avoid the computation of \( \kappa \) at each iteration (issue 1) which costs as much as the full gradient algorithm, while following closely the changes of the dual residue \( \kappa \). We reset the adaptive sampling probability after every epoch of length \( n \).

**Parameter** \( m \) The setting of parameter \( m \) in AdaSDCA+ directly affects the performance of the algorithm. If \( m \) is too large, the probability of sampling the same coordinate
Algorithm 2 AdaSDCA+

Parameter a number \( m > 1 \),
Initialization Choose \( \alpha^0 \in \mathbb{R}^n \), set \( \bar{\alpha}^0 = \frac{1}{\lambda m} A \alpha^0 \)
for \( t \geq 0 \) do
  Primal update: \( w^t = \nabla g^* (\alpha^t) \)
  Set: \( \alpha^{t+1} = \alpha^t \)
  if \( \text{mod} (t, n) == 0 \) then
    **Option I**: Adaptive sampling:
    Compute: \( \kappa_i^t = \alpha_i^t + \nabla \phi_i(A_i^T w^t), \forall i \in [n] \)
    Set: \( p_i^t \sim \kappa_i^t / \sqrt{v_i + n \lambda \gamma}, \forall i \in [n] \)
    **Option II**: Optimal importance sampling:
    Set: \( p_i^t \sim (v_i + n \lambda \gamma), \forall i \in [n] \)
  end if
  Generate random \( i_t \in [n] \) according to \( p^t \)
  Compute:
  \[
  \Delta \alpha^t_{i_t} = \arg \max_{\Delta \in \mathbb{R}} \{-\phi_{i_t}^* \left( (\alpha^t_{i_t} + \Delta) \right) - A_i^T w^t \Delta - \frac{v_i}{2 m n} \Delta^2 \}
  \]
  Dual update: \( \alpha^{t+1}_{i_t} = \alpha^t_{i_t} + \Delta \alpha^t_{i_t} \)
  Average update: \( \bar{\alpha}^t = \bar{\alpha}^{t+1} + \frac{\Delta \alpha^t_{i_t}}{m} A_{i_t} \)
  Probability update:
  \( p_{i_t}^{t+1} \sim p_{i_t}^t / m, \quad p_{j}^{t+1} \sim p_{j}^t, \forall j \neq i \)
end for
Output: \( w^t, \bar{\alpha}^t \)

Table 1. One epoch computational cost of different algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cost of an Epoch</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDCA&amp; QUARTZ(uniform)</td>
<td>( O(nnz) )</td>
</tr>
<tr>
<td>IPROX-SDCA</td>
<td>( O(nnz + n \log(n)) )</td>
</tr>
<tr>
<td>ADASDCA</td>
<td>( O(n \cdot nnz) )</td>
</tr>
<tr>
<td>ADA SDCA+</td>
<td>( O(nnz + n \log(n)) )</td>
</tr>
</tbody>
</table>

Table 1 shows the total computational cost of one epoch for different algorithms. The computational cost includes the cost of updating the primal variable, the dual variable, and the sampling probability.

6. Numerical Experiments

In this section we present results of numerical experiments. Additional experiments can be found in the supplementary materials.

6.1. Loss functions

We test AdaSDCA and AdaSDCA+, SDCA, and IProx-SDCA for two different types of loss functions \( \{\phi_i\}_{i=1}^n \): quadratic loss and smoothed Hinge loss. Let \( y \in \mathbb{R}^n \) be the vector of labels. The quadratic loss is given by

\[
\phi_i(x) = \frac{1}{2\gamma} (x - y_i)^2
\]

and the smoothed Hinge loss is:

\[
\phi_i(x) = \begin{cases} 
0 & y_i x \geq 1 \\
1 - y_i x - \gamma/2 & y_i x \leq 1 - \gamma \\
\frac{(1-y_i x)^2}{2\gamma} & \text{otherwise,}
\end{cases}
\]

In both cases we use \( L_2 \)-regularizer, i.e.,

\[
g(w) = \frac{1}{2} \| w \|^2.
\]

Quadratic loss functions appear usually in regression problems, and smoothed Hinge loss can be found in linear support vector machine (SVM) problems (Shalev-Shwartz & Zhang, 2013a).
Table 2. Dimensions and nonzeros of the datasets

<table>
<thead>
<tr>
<th>DATASET</th>
<th>d</th>
<th>n</th>
<th>nnz / (nd)</th>
</tr>
</thead>
<tbody>
<tr>
<td>w8a</td>
<td>300</td>
<td>49,749</td>
<td>3.9%</td>
</tr>
<tr>
<td>DOROTHEA</td>
<td>100,000</td>
<td>800</td>
<td>0.9%</td>
</tr>
<tr>
<td>MUSHROOMS</td>
<td>112</td>
<td>8,124</td>
<td>18.8%</td>
</tr>
<tr>
<td>COV1</td>
<td>54</td>
<td>581,012</td>
<td>22%</td>
</tr>
<tr>
<td>IJCNN1</td>
<td>22</td>
<td>49,990</td>
<td>41%</td>
</tr>
</tbody>
</table>

6.2. Numerical results

We used 5 different datasets: w8a, dorothea, mushrooms, cov1 and ijcnn1 (see Table 2).

In all our experiments we used $\gamma = 1$ and $\lambda = 1/n$.

**AdaSDCA** The results of the theory developed in Section 4 can be observed on Figure 1. AdaSDCA needs the least amount of iterations to converge, confirming the theoretical result.

**AdaSDCA+ V.S. others** We can observe on Figure 6 and Figure 7, that both options of AdaSDCA+ outperforms SDCA and IProx-SDCA, in terms of number of iterations, for quadratic loss functions and for smoothed Hinge loss functions. One can observe similar results in terms of time through Figure 2 to Figure 5.

**Option I V.S. Option II** Despite the fact that Option I is not theoretically supported for smoothed hinge loss, it still converges faster than Option II on every dataset and for every loss function. The biggest difference can be observed on Figure 5, where Option I converges to the machine precision in just 15 seconds.

**Different choices of m** To show the impact of different choices of m on the performance of AdaSDCA+, in Figure 8 and Figure 9 we compare the results of the two options of AdaSDCA+ using different m equal to 2, 10 and 50. It is hard to draw a clear conclusion here because clearly the optimal m shall depend on the dataset and the problem type.

**Remark 14.** Recently there has been several work on applying Nesterov’s acceleration technique to SDCA so that the iteration complexity can be improved from $\tilde{O}(n + \frac{1}{\lambda \gamma})$ to $\tilde{O}(n + \sqrt{\frac{n}{\lambda \gamma}})$, see (Shalev-Shwartz & Zhang, 2013a; 2015; Zhang & Xiao, 2015; Lin et al., 2014). In all our numerical experiments, the condition number $1/\lambda \gamma$ equals n, in which case SDCA and accelerated SDCA are comparable. It is important to notice that the adaptive sampling technique developed in this paper is an independent speedup trick to the acceleration technique used in the above cited papers. Whether the two approaches can be combined together is left for future work.

 Acknowledgements

The authors thank Jakub Konečný for his valuable suggestions.

Figure 1. w8a dataset $d = 300$, $n = 49749$, Quadratic loss with $L_2$ regularizer, comparing number of iterations with known algorithms

Figure 2. dorothea dataset $d = 100000$, $n = 800$, Quadratic loss with $L_2$ regularizer, comparing real time with known algorithms

Figure 3. ijcnn1 dataset $d = 22$, $n = 49990$, Quadratic loss with $L_2$ regularizer, comparing real time with known algorithms
Figure 4. mushrooms dataset $d = 112$, $n = 8124$, Smooth Hinge loss with $L_2$ regularizer, comparing real time with known algorithms

Figure 5. cov1 dataset $d = 54$, $n = 581012$, Smooth Hinge loss with $L_2$ regularizer, comparing real time with known algorithms

Figure 6. ijcnn1 dataset $d = 22$, $n = 49990$, Quadratic loss with $L_2$ regularizer, comparing number of iterations with known algorithms

Figure 7. w8a dataset $d = 300$, $n = 49749$, Smooth Hinge loss with $L_2$ regularizer, comparing number of iterations with known algorithms

Figure 8. cov1 dataset $d = 54$, $n = 581012$, Quadratic loss with $L_2$ regularizer, comparison of different choices of the constant $m$

Figure 9. mushrooms dataset $d = 112$, $n = 8124$, Smooth Hinge loss with $L_2$ regularizer, comparison of different choices of the constant $m$
Stochastic Dual Coordinate Ascent with Adaptive Probabilities

References


Zhao, Peilin and Zhang, Tong. Stochastic optimization with importance sampling. ICML, 2015.
