Stochastic Dual Coordinate Ascent with Adaptive Probabilities

Dominik Csiba CDOMINIK@GMAIL.COM
Zheng Qu ZHENG.QU@ED.AC.UK
Peter Richtárik PETER.RICHTARIK@ED.AC.UK
University of Edinburgh

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 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:

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

In the context of supervised learning, w is a linear predictor, $A_1,\dots,A_n\in\mathbb{R}^d$ are samples, $\phi_1,\dots,\phi_n:\mathbb{R}^d\to\mathbb{R}$ 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 $1/\gamma$:

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

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

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

for all $w_1,w_2\in\operatorname{dom}g$, $0\le\alpha\le1$ 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, 2014). 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):

$$\max_{\alpha=(\alpha_1,\dots,\alpha_n)\in\mathbb{R}^n}\left[D(\alpha)\stackrel{\text{def}}{=}-f(\alpha)-\psi(\alpha)\right], \qquad (2)$$

where functions f and ψ are defined by

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

$$\psi(\alpha)\stackrel{\text{def}}{=}\frac{1}{n}\sum_{i=1}^{n}\phi_i^*(-\alpha_i), \qquad (4)$$

and g^* and ϕ_i^* are the convex conjugates[1] of g and ϕ_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 α_i, and performs its update, usually via closed-form for-

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

mula – this strategy is know as randomized coordinate descent. Methods based on updating randomly selected dual variables enjoy, in our setting, a linear convergence rate (Shalev-Shwartz & Zhang, 2013b; 2012; Takáč et al., 2013; Shalev-Shwartz & Zhang, 2013a; Zhao & Zhang, 2014; Qu et al., 2014). These methods have attracted considerable attention in the past few years, and include SCD (Shalev-Shwartz & Tewari, 2011), RCDM (Nesterov, 2012), UCDC (Richtárik & Takáč, 2014), ICD (Tappenden et al., 2013), PCDM (Richtárik & Takáč, 2012), SPCDM (Fercoq & Richtárik, 2013), SPDC (Zhang & Xiao, 2014), APCG (Lin et al., 2014), RCD (Necoara & Patrascu, 2014), APPROX (Fercoq & Richtárik, 2013), QUARTZ (Qu et al., 2014) and ALPHA (Qu & Richtárik, 2014). Recent advances on mini-batch and distributed variants can be found in (Liu & Wright, 2014), (Zhao et al., 2014b), (Richtárik & Takáč, 2013a), (Fercoq et al., 2014), (Trofimov & Genkin, 2014), (Jaggi et al., 2014), (Mareček et al., 2014) and (Mahajan et al., 2014). Other related work includes (Nemirovski et al., 2009; Duchi et al., 2011; Agarwal & Bottou, 2014; Zhao et al., 2014a; Fountoulakis & Tappenden, 2014; Tappenden et al., 2014). We also point to (Wright, 2014) for a review on coordinate descent algorithms.

Selection Probabilities. Naturally, both the theoretical convergence rate and practical performance of randomized coordinate descent methods depends on the probability distribution governing the choice of individual coordinates. While most existing work assumes uniform distribution, it was shown by Richtárik & Takáč (2014); Necoara et al. (2012); Zhao & Zhang (2014) that coordinate descent works for an arbitrary fixed probability distribution over individual coordinates and even subsets of coordinates (Richtárik & Takáč, 2013b; Qu et al., 2014; Qu & Richtárik, 2014; Qu & Richtárik, 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 (Richtárik & Takáč, 2014; Qu & Richtárik, 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 (Richtárik & Takáč, 2013b; Zhao & Zhang, 2014; Qu et al., 2014; Qu & Richtárik, 2014; Qu & Richtárik, 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., 2011), 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) and describe the intuition behind it. In Section 4 we provide convergence analysis. In Section 5 we introduce Algorithm 2: an variant of 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 appendix.

2. Contributions

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 the dual coordinates.

Complexity analysis. We provide a convergence rate analysis for the first method, showing that **AdaSDCA enjoys better rate than the best known rate for SDCA with a fixed sampling** (Zhao & Zhang, 2014; Qu et al., 2014). The probabilities are proportional to a certain measure of dual suboptimality associated with each variable.

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, 2014). We support this with computational experiments in Section 6.

Outline: In Section 2 we summarize the contributions of our work. In Section 3 we describe our first, theoretical methods (AdaSDCA) and describe the intuition behind it. In Section 4 we provide convergence analysis. In Section 5 we introduce AdaSDCA+: a variant of AdaSDCA 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 appendix.

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*:

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

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

where A is the d-by-n matrix with columns $A_1, \dots, A_n$.

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

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

Note, that $\kappa_i^t = 0$ if and only if α_i satisfies (5). This motivates the design of AdaSDCA (Algorithm 1) as follows: whenever $|\kappa_i^t|$ is large, the ith dual coordinate α_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 κ^t if for all $i \in [n]$ we have

$$\kappa_i^t \neq 0 \quad \Rightarrow \quad p_i^t > 0.$$

Alternatively, p^t is coherent with k^t if for

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

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

Algorithm 1 AdaSDCA

Init: $v_i = A_i^\top A_i$ for $i \in [n]$; $\alpha^0 \in \mathbb{R}^n$; $\bar{\alpha}^0 = \frac{1}{\lambda n} A\alpha^0$
for $t \geq 0$ **do**
  Primal update: $w^t = \nabla g^* (\bar{\alpha}^t)$
  Set: $\alpha^{t+1} = \alpha^t$
  Compute residue κ^t: $\kappa_i^t = \alpha_i^t + \nabla \phi_i(A_i^\top w^t), \forall i \in [n]$
  Compute probability distribution p^t coherent with κ^t
  Generate random $i_t \in [n]$ according to p^t
  Compute:

$$\Delta\alpha_{i_t}^t = \arg\max_{\Delta \in \mathbb{R}} \left\{ -\phi_{i_t}^*(-(\alpha_{i_t}^t + \Delta)) - A_{i_t}^\top w^t \Delta - \tfrac{v_{i_t}}{2\lambda n} |\Delta|^2 \right\}$$

  Dual update: $\alpha_{i_t}^{t+1} = \alpha_{i_t}^t + \Delta\alpha_{i_t}^t$
  Average update: $\bar{\alpha}^t = \bar{\alpha}^t + \frac{\Delta\alpha_{i_t}}{\lambda n} A_{i_t}$
end for
Output: w^t, α^t

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 κ.

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 $\mathbb{E}_t$ the expectation with respect to the random index $i_t \in [n]$ generated at iteration t.

Lemma 3. *Consider the AdaSDCA algorithm during iteration $t \geq 0$ and assume that p^t is coherent with κ^t. Then*

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

for arbitrary

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

Proof. Lemma 3 is proved similarly to Lemma 2 in (Zhao & Zhang, 2014), but in a slightly more general setting. For completeness, we provide the proof in the appendix. □

Lemma 3 plays a key role in the analysis of stochastic dual coordinate methods (Shalev-Shwartz & Zhang, 2013b; Zhao & Zhang, 2014; Shalev-Shwartz & Zhang, 2013a). Indeed, if the right-hand side of (8) is positive, then the primal dual error $P(w^t) - D(\alpha^t)$ can be bounded by the expected dual ascent $\mathbb{E}_t[D(\alpha^{t+1}) - D(\alpha^t)]$ times $1/\theta$, which yields the contraction of the dual error at the rate of $1 - \theta$ (see Theorem 7). In order to make the right-hand side of (8) positive we can take any θ smaller than $\theta(\kappa^t, p^t)$ where the function $\theta(\cdot, \cdot) : \mathbb{R}_+^n \times \mathbb{R}_+^n \to \mathbb{R}$ is defined by:

$$\theta(\kappa, p) \equiv \frac{n\lambda\gamma \sum_{i:\kappa_i \neq 0} |\kappa_i|^2}{\sum_{i:\kappa_i \neq 0} p_i^{-1} |\kappa_i|^2 (v_i + n\lambda\gamma)}. \tag{10}$$

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:

$$\begin{aligned} \max_{p \in \mathbb{R}_+^n} \quad & \theta(\kappa^t, p) \\ \text{s.t.} \quad & \sum_{i=1}^n p_i = 1 \\ & \theta(\kappa^t, p) \leq \min_{i:\kappa_i^t \neq 0} p_i \end{aligned} \tag{11}$$

A feasible solution to (11) is the *importance sampling* (also known as optimal serial sampling) p^* defined by:

$$p_i^* \overset{\text{def}}{=} \frac{v_i + n\lambda\gamma}{\sum_{j=1}^n (v_j + n\lambda\gamma)}, \quad \forall i \in [n], \tag{12}$$

which was proposed in (Zhao & Zhang, 2014) 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^t is independent of the residue κ^t, the computation of κ^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 $\theta(\kappa^t, p) \leq \min_{i:\kappa_i^t \neq 0} p_i$, we obtain an explicit optimal solution.

Lemma 4. *The optimal solution $p^*(\kappa^t)$ of*

$$\begin{aligned} \max_{p \in \mathbb{R}^n_+} \quad & \theta(\kappa^t, p) \\ \text{s.t.} \quad & \sum_{i=1}^n p_i = 1 \end{aligned} \tag{13}$$

is:

$$(p^*(\kappa^t))_i = \frac{|\kappa_i^t|\sqrt{v_i + n\lambda\gamma}}{\sum_{j=1}^n |\kappa_j^t|\sqrt{v_j + n\lambda\gamma}}, \quad \forall i \in [n]. \tag{14}$$

Proof. The proof is deferred to the appendix. □

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

If we let $p^t = p^*(\kappa^t)$ and $\theta = \theta(\kappa^t, p^t)$, the constraint (9) may not be sastified, 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 $\{\phi_i\}_i$ are quadratic.

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

The proof is deferred to Appendix.

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 $\{\phi_i\}_{i=1}^n$ are all quadratic.

4.1. General loss functions

We derive the convergence result from Lemma 3.

Proposition 6. *Let $t \geq 0$. If $\min_{i \in I_t} p_i^t > 0$ and $\theta(\kappa^t, p^t) \leq \min_{i \in I_t} p_i^t$, then*

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

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 $t \geq 0$, $\min_{i \in I_t} p_i^t > 0$ and $\theta(\kappa^t, p^t) \leq \min_{i \in I_t} p_i^t$, then*

$$\mathbb{E}[P(w^t) - D(\alpha^t)] \leq \frac{1}{\tilde{\theta}_t} \prod_{k=0}^t (1 - \tilde{\theta}_k)\left(D(\alpha^*) - D(\alpha^0)\right), \tag{15}$$

for all $t \geq 0$ where

$$\tilde{\theta}_t \overset{def}{=} \frac{\mathbb{E}[\theta(\kappa^t, p^t)(P(w^t) - D(\alpha^t))]}{\mathbb{E}[P(w^t) - D(\alpha^t)]}. \tag{16}$$

Proof. By Proposition 6, we know that

$$\begin{aligned} \mathbb{E}[D(\alpha^{t+1}) - D(\alpha^t)] &\geq \mathbb{E}[\theta(\kappa^t, p^t)(P(w^t) - D(\alpha^t))] \\ &\overset{(16)}{=} \tilde{\theta}_t\, \mathbb{E}[P(w^t) - D(\alpha^t)] \\ &\geq \tilde{\theta}_t\, \mathbb{E}[D(\alpha^*) - D(\alpha^t)], \end{aligned} \tag{17}$$

whence

$$\mathbb{E}[D(\alpha^*) - D(\alpha^{t+1})] \leq (1 - \tilde{\theta}_t)\, \mathbb{E}[D(\alpha^*) - D(\alpha^t)].$$

Therefore,

$$\mathbb{E}[D(\alpha^*) - D(\alpha^t)] \leq \prod_{k=0}^t (1 - \tilde{\theta}_k)\left(D(\alpha^*) - D(\alpha^0)\right).$$

By plugging the last bound into (17) we get the bound on the primal dual error:

$$\begin{aligned} \mathbb{E}[P(w^t) - D(\alpha^t)] &\leq \frac{1}{\tilde{\theta}_t}\, \mathbb{E}[D(\alpha^{t+1}) - D(\alpha^t)] \\ &\leq \frac{1}{\tilde{\theta}_t}\, \mathbb{E}[D(\alpha^*) - D(\alpha^t)] \\ &\leq \frac{1}{\tilde{\theta}_t} \prod_{k=0}^t (1 - \tilde{\theta}_k)\left(D(\alpha^*) - D(\alpha^0)\right). \end{aligned}$$ □

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, 2014). The convergence theory established for IProx-SDCA in (Zhao & Zhang, 2014), which can also be derived as a direct corollary of our Theorem 7, is stated as follows.

Theorem 8 ((Zhao & Zhang, 2014)). *Consider AdaSDCA with $p^t = p^*$ defined in (12) for all $t \geq 0$. Then*

$$\mathbb{E}[P(w^t) - D(\alpha^t)] \leq \frac{1}{\theta_*}(1-\theta_*)^t \left(D(\alpha^*) - D(\alpha^0)\right),$$

where

$$\theta_* = \frac{n\lambda\gamma}{\sum_{i=1}^n (v_i + \lambda\gamma n)}.$$

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 $\min_{i \in I_t} p_i^t > 0$, then*

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

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 $t \geq 0$, $\min_{i \in I_t} p_i^t > 0$, 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 and $\tilde{\theta}_t \geq \theta_*$ for all $t \geq 0$.*

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, 2014). However, there are two main implementation issues of AdaSDCA:

1. The update of the dual residue κ^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

Algorithm 2 AdaSDCA+

Parameter a number $m > 1$
Initialization Choose $\alpha^0 \in \mathbb{R}^n$, set $\bar{\alpha}^0 = \frac{1}{\lambda n} A\alpha^0$
for $t \geq 0$ **do**
  Primal update: $w^t = \nabla g^*(\bar{\alpha}^t)$
  Set: $\alpha^{t+1} = \alpha^t$
  if $\mod(t, n) == 0$ **then**
    Option I: Adaptive probability
      Compute: $\kappa_i^t = \alpha_i^t + \nabla\phi_i(A_i^\top w^t),\ \forall i \in [n]$
      Set: $p_i^t \sim |\kappa_i^t|\sqrt{v_i + n\lambda\gamma},\ \forall i \in [n]$
    Option II: Optimal Importance probability
      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:

$$\Delta\alpha_{i_t}^t = \arg\max_{\Delta \in \mathbb{R}} \left\{-\phi_{i_t}^*(-(\alpha_{i_t}^t + \Delta)) - A_{i_t}^\top w^t \Delta - \frac{v_{i_t}}{2\lambda n}|\Delta|^2\right\}$$

  Dual update: $\alpha_{i_t}^{t+1} = \alpha_{i_t}^t + \Delta\alpha_{i_t}^t$
  Average update: $\bar{\alpha}^t = \bar{\alpha}^t + \frac{\Delta\alpha_{i_t}}{\lambda n} A_{i_t}$
  Probability update:

$$p_{i_t}^{t+1} \sim p_{i_t}^t/m,\ \ p_j^{t+1} \sim p_j^t, \forall j \neq i_t$$

end for
Output: w^t, α^t

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 α_i is updated, the residue κ_i naturally decreases. We then decrease also the probability that i is chosen in the next iteration, by setting p^{t+1} to be proportional to $(p_1^t, \ldots p_{i-1}^t, p_i^t/m, p_{i+1}^t, \ldots, p_n^t)$. By doing this

we avoid the computation of κ at each iteration (issue 1) which costs as much as the full gradient algorithm, while following closely the changes of the dual residue κ. 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 twice during an epoch will be very small. This will result in a random permutation through all coordinates every epoch. On the other hand, for m too small the coordinates having larger probabilities at the beginning of an epoch could be sampled more often than it should, even after their corresponding dual residues become sufficiently small. We don't have a definitive rule on the choice of m and we leave this to future work. Experiments with different choices of m can be found in Section 6.

Option I & Option II At the beginning of each epoch, one can choose between two options for resetting the sampling probability. Option I corresponds to the optimal solution of (13), given by the closed form (14). Option II is the optimal serial sampling probability (12), the same as the one used in IProx-SDCA (Zhao & Zhang, 2014). However, AdaSDCA+ differs significantly with IProx-SDCA since we also update iteratively the sampling probability, which as we show through numerical experiments yields a faster convergence than IProx-SDCA.

5.2. Computational cost

Sampling and probability update During the algorithm we sample $i \in [n]$ from non-uniform probability distribution p^t, which changes at each iteration. This process can be done efficiently using the Random Counters algorithm introduced in Section 6.2 of (Nesterov, 2012), which takes $O(n\log(n))$ operations to create the probability tree and $O(\log(n))$ operations to sample from the distribution or change one of the probabilities.

Total computational cost We can compute the computational cost of one epoch. At the beginning of an epoch, we need $O(\text{nnz})$ operations to calculate the dual residue κ. Then we create a probability tree using $O(n\log(n))$ operations. At each iteration we need $O(\log(n))$ operations to sample a coordinate, $O(\text{nnz}/n)$ operations to calculate the update to α and a further $O(\log(n))$ operations to update the probability tree. As a result an epoch needs $O(\text{nnz}+n\log(n))$ operations. For comparison purpose we list in Table 1 the one epoch computational cost of comparable algorithms.

6. Numerical Experiments

In this section we present results of numerical experiments.

Table 1. One epoch computational cost of different algorithms

ALGORITHM	COST OF AN EPOCH
SDCA& QUARTZ(UNIFORM)	$O(\text{nnz})$
IPROX-SDCA	$O(\text{nnz}+n\log(n))$
ADASDCA	$O(n\cdot\text{nnz})$
ADASDCA+	$O(\text{nnz}+n\log(n))$

Table 2. Dimensions and nonzeros of the datasets

DATASET	d	n	$\text{nnz}/(nd)$
W8A	300	49,749	3.9%
DOROTHEA	100,000	800	0.9%
MUSHROOMS	112	8,124	18.8%
COV1	54	581,012	22%
IJCNN1	22	49,990	41%

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).

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 through Figure 1 to Figure 4. AdaSDCA needs the least amount of iterations to converge, confirming the theoretical result.

AdaSDCA+ V.S. others We can observe through Figure 15 to 24, 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 5 to Figure 14.

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 13, 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 Figures 25 to 33 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.

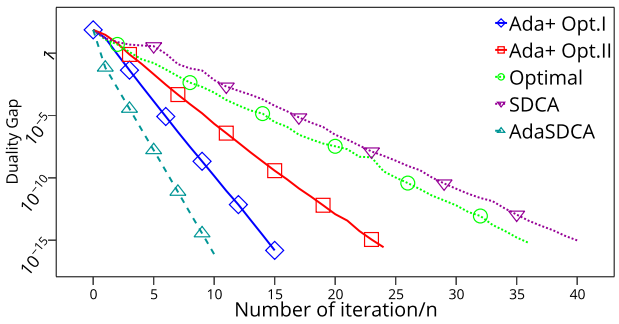

Figure 1. w8a dataset d = 300, n = 49749, Quadratic loss with L_2 regularizer, comparing number of iterations with known algorithms

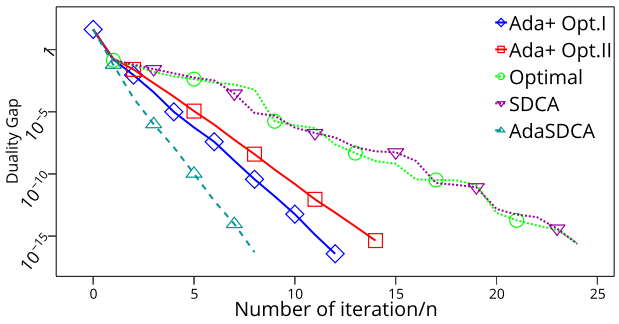

Figure 2. dorothea dataset d = 100000, n = 800, Quadratic loss with L_2 regularizer, comparing number of iterations with known algorithms

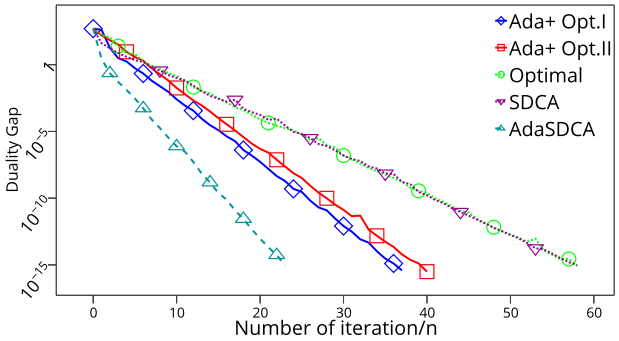

Figure 3. mushrooms dataset d = 112, n = 8124, Quadratic loss with L_2 regularizer, comparing number of iterations with known algorithms

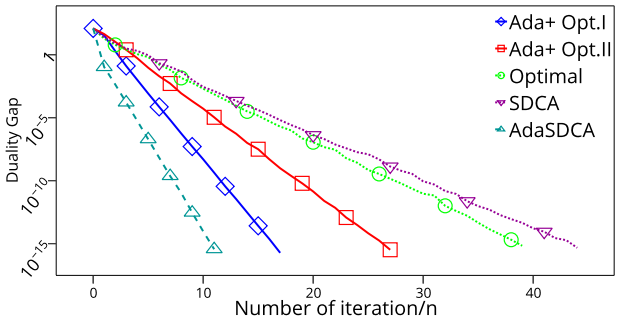

Figure 4. ijcnn1 dataset d = 22, n = 49990, Quadratic loss with L_2 regularizer, comparing number of iterations with known algorithms

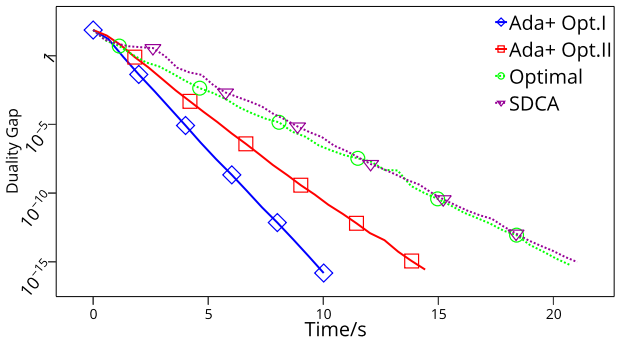

Figure 5. w8a dataset $d = 300$, $n = 49749$, Quadratic loss with L_2 regularizer, comparing real time with known algorithms

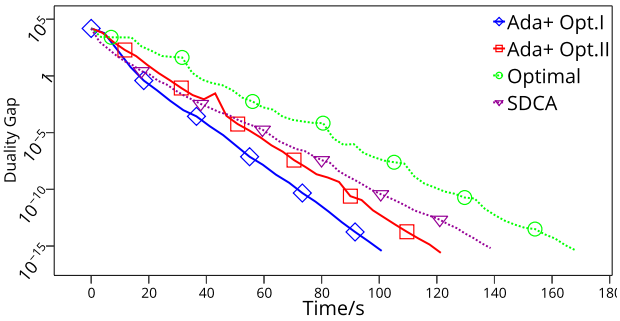

Figure 8. cov1 dataset $d = 54$, $n = 581012$, Quadratic loss with L_2 regularizer, comparing real time with known algorithms

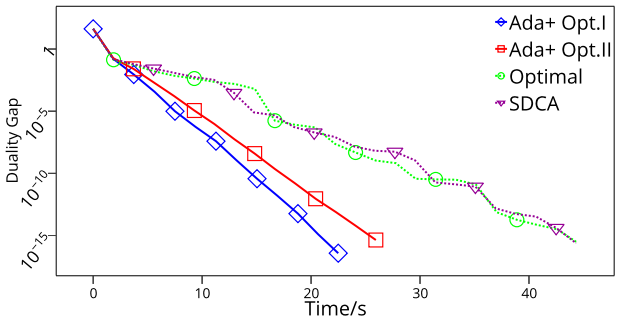

Figure 6. dorothea dataset $d = 100000$, $n = 800$, Quadratic loss with L_2 regularizer, comparing real time with known algorithms

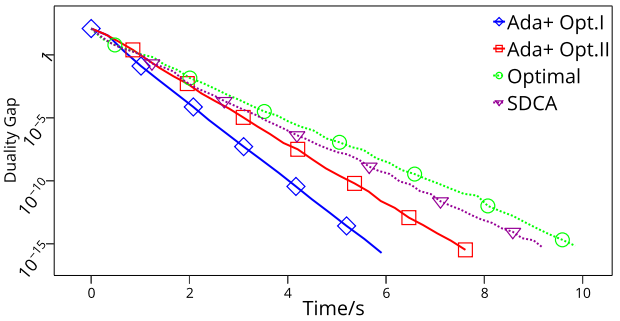

Figure 9. ijcnn1 dataset $d = 22$, $n = 49990$, Quadratic loss with L_2 regularizer, comparing real time with known algorithms

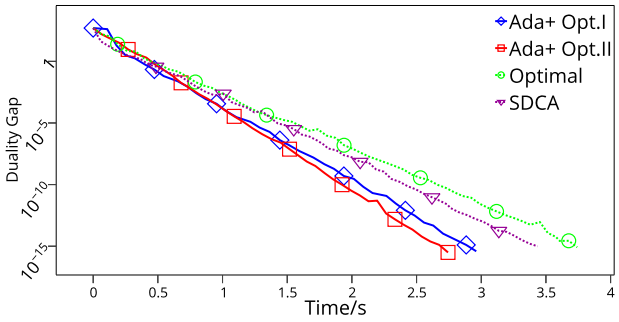

Figure 7. mushrooms dataset $d = 112$, $n = 8124$, Quadratic loss with L_2 regularizer, comparing real time with known algorithms

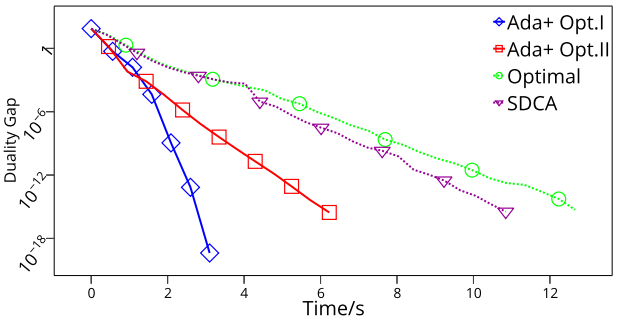

Figure 10. w8a dataset $d = 300$, $n = 49749$, Smooth Hinge loss with L_2 regularizer, comparing real time with known algorithms

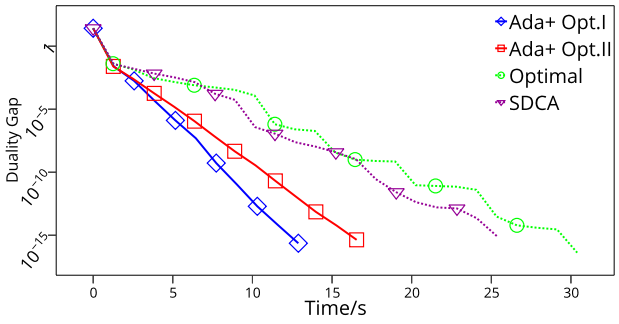

Figure 11. dorothea dataset $d = 100000$, $n = 800$, Smooth Hinge loss with L_2 regularizer, comparing real time with known algorithms

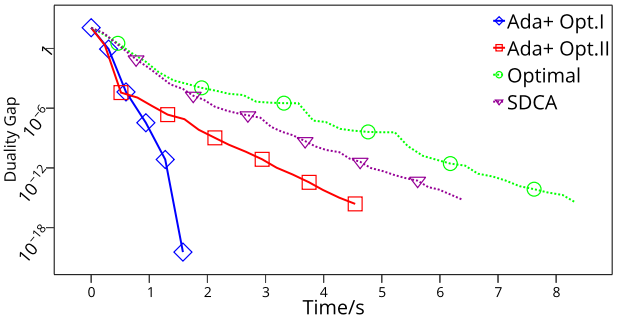

Figure 14. ijcnn1 dataset $d = 22$, $n = 49990$, Smooth Hinge loss with L_2 regularizer, comparing real time with known algorithms

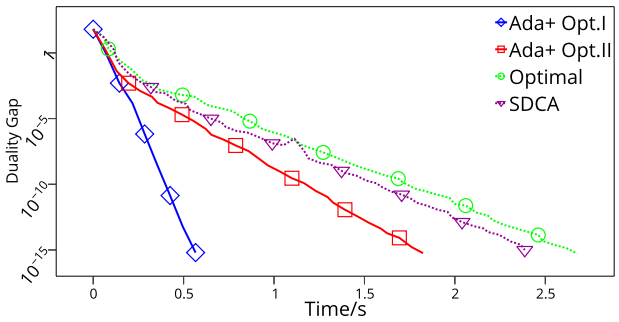

Figure 12. mushrooms dataset $d = 112$, $n = 8124$, Smooth Hinge loss with L_2 regularizer, comparing real time with known algorithms

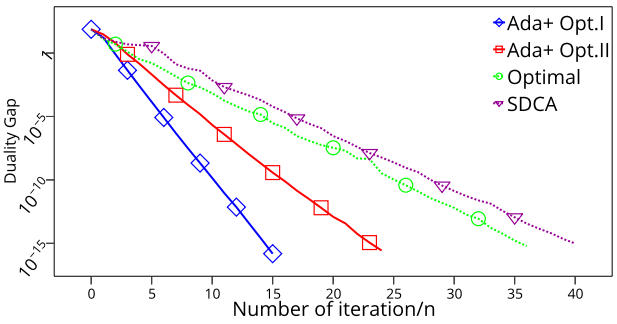

Figure 15. w8a dataset $d = 300$, $n = 49749$, Quadratic loss with L_2 regularizer, comparing number of iterations with known algorithms

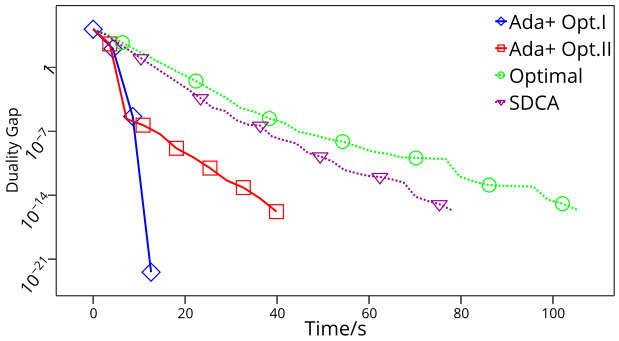

Figure 13. cov1 dataset $d = 54$, $n = 581012$, Smooth Hinge loss with L_2 regularizer, comparing real time with known algorithms

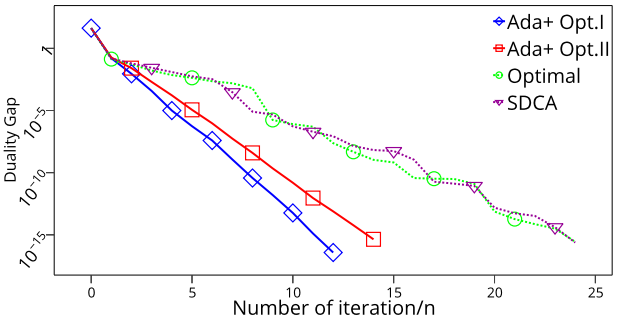

Figure 16. dorothea dataset $d = 100000$, $n = 800$, Quadratic loss with L_2 regularizer, comparing number of iterations with known algorithms

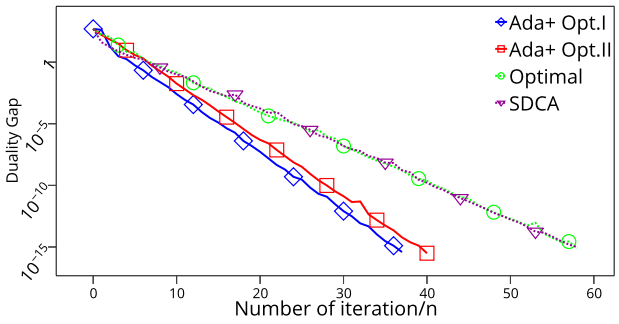

Figure 17. mushrooms dataset $d = 112$, $n = 8124$, Quadratic loss with L_2 regularizer, comparing number of iterations with known algorithms

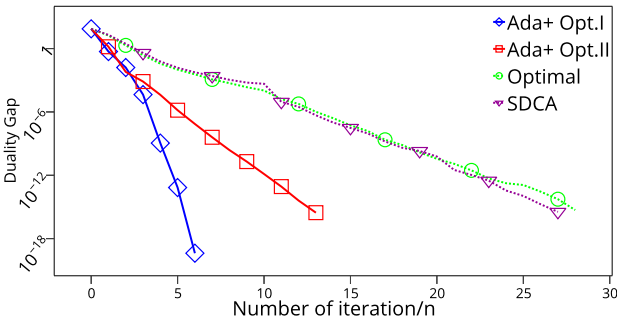

Figure 20. w8a dataset $d = 300$, $n = 49749$, Smooth Hinge loss with L_2 regularizer, comparing number of iterations with known algorithms

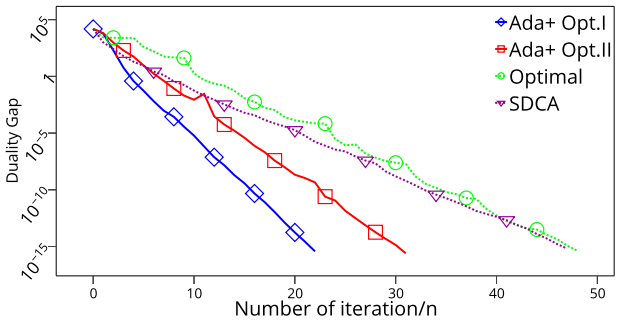

Figure 18. cov1 dataset $d = 54$, $n = 581012$, Quadratic loss with L_2 regularizer, comparing number of iterations with known algorithms

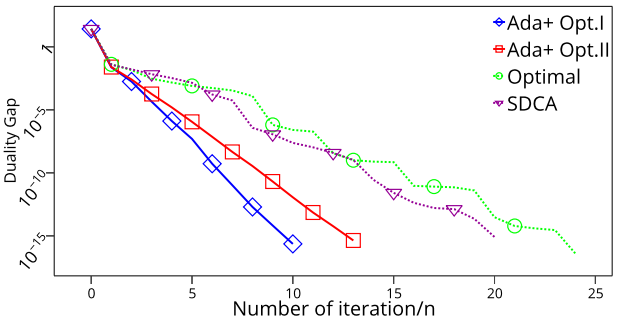

Figure 21. dorothea dataset $d = 100000$, $n = 800$, Smooth Hinge loss with L_2 regularizer, comparing number of iterations with known algorithms

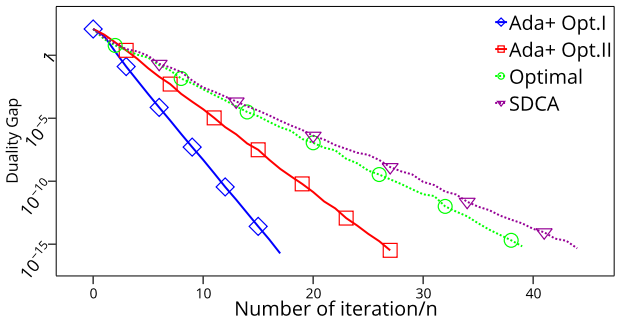

Figure 19. ijcnn1 dataset $d = 22$, $n = 49990$, Quadratic loss with L_2 regularizer, comparing number of iterations with known algorithms

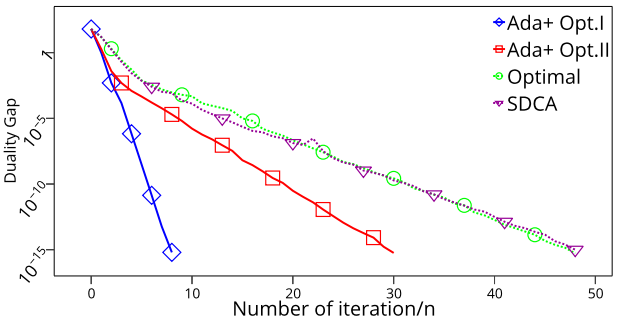

Figure 22. mushrooms dataset $d = 112$, $n = 8124$, Smooth Hinge loss with L_2 regularizer, comparing number of iterations with known algorithms

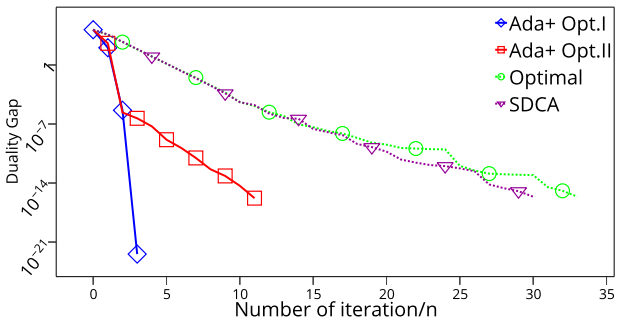

Figure 23. cov1 dataset $d = 54$, $n = 581012$, Smooth Hinge loss with L_2 regularizer, comparing number of iterations with known algorithms

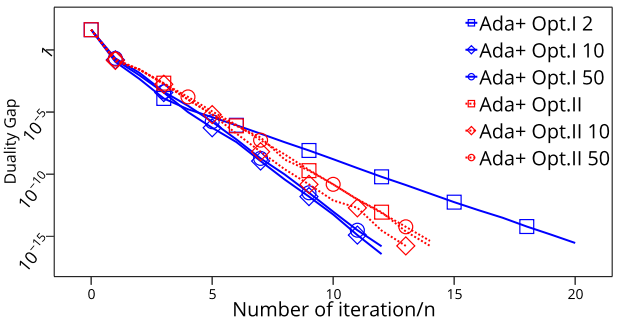

Figure 26. dorothea dataset $d = 100000$, $n = 800$, Quadratic loss with L_2 regularizer, comparison of different choices of the constant m

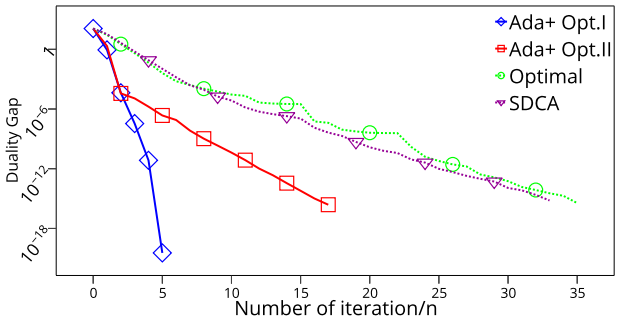

Figure 24. ijcnn1 dataset $d = 22$, $n = 49990$, Smooth Hinge loss with L_2 regularizer, comparing number of iterations with known algorithms

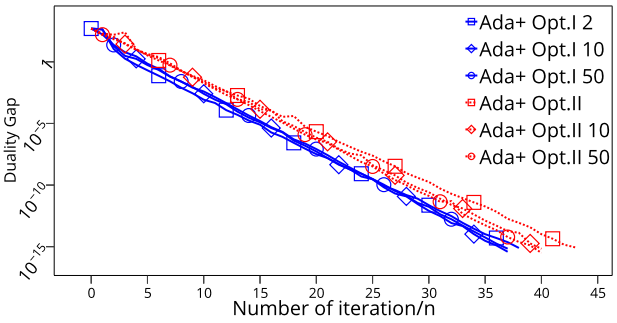

Figure 27. mushrooms dataset $d = 112$, $n = 8124$, Quadratic loss with L_2 regularizer, comparison of different choices of the constant m

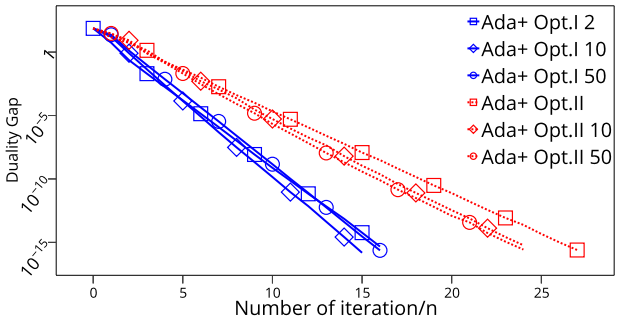

Figure 25. w8a dataset $d = 300$, $n = 49749$, Quadratic loss with L_2 regularizer, comparison of different choices of the constant m

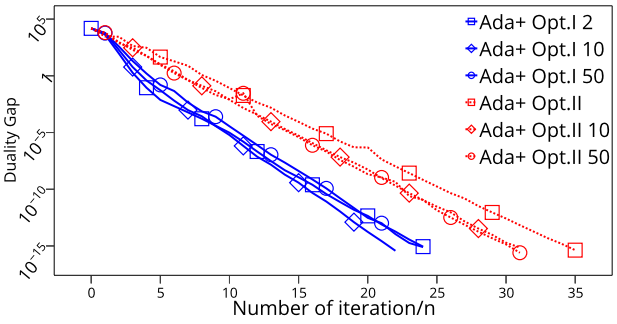

Figure 28. cov1 dataset $d = 54$, $n = 581012$, Quadratic loss with L_2 regularizer, comparison of different choices of the constant m

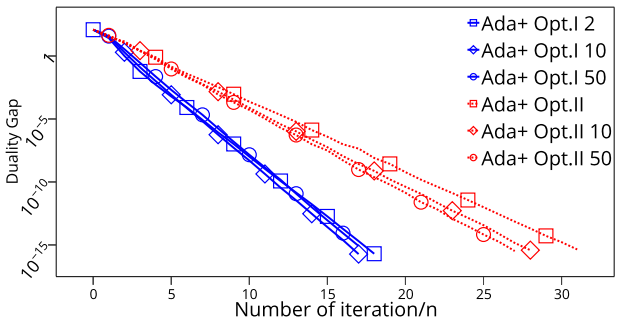

Figure 29. ijcnn1 dataset $d = 22$, $n = 49990$, Quadratic loss with L_2 regularizer, comparison of different choices of the constant m

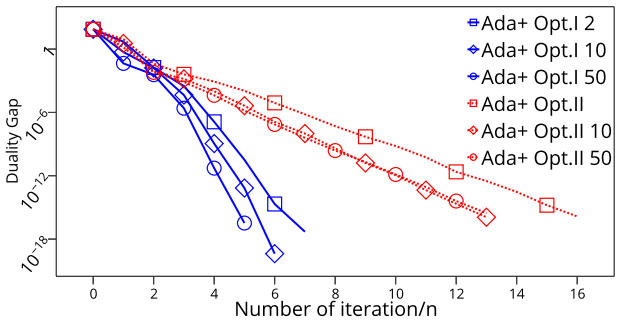

Figure 30. w8a dataset $d = 300$, $n = 49749$, Smooth Hinge loss with L_2 regularizer, comparison of different choices of the constant m

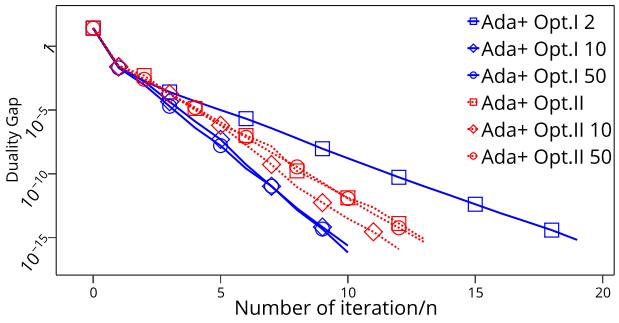

Figure 31. dorothea dataset $d = 100000$, $n = 800$, Smooth Hinge loss with L_2 regularizer, comparison of different choices of the constant m

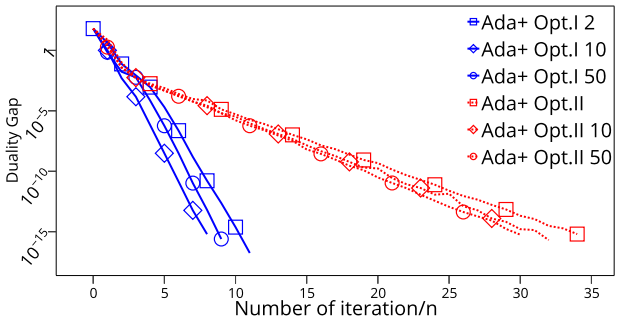

Figure 32. mushrooms dataset $d = 112$, $n = 8124$, Smooth Hinge loss with L_2 regularizer, comparison of different choices of the constant m

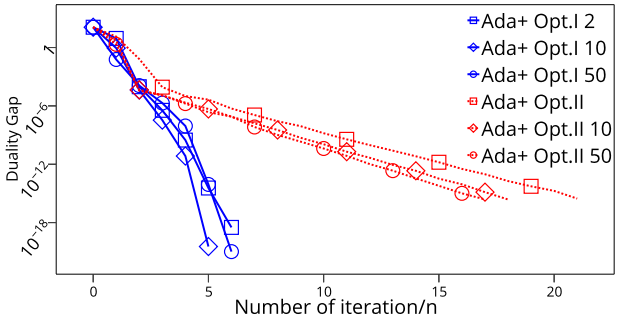

Figure 33. ijcnn1 dataset $d = 22$, $n = 49990$, Smooth Hinge loss with L_2 regularizer, comparison of different choices of the constant m

References

Agarwal, Alekh and Bottou, Leon. A lower bound for the optimization of finite sums. *arXiv:1410.0723*, 2014.

Banks-Watson, Alexander. New classes of coordinate descent methods. Master's thesis, University of Edinburgh, 2012.

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *arXiv:1407.0202*, 2014.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(1):2121–2159, 2011.

Fercoq, Olivier and Richtárik, Peter. Accelerated, parallel and proximal coordinate descent. *SIAM Journal on Optimization (after minor revision), arXiv:1312.5799*, 2013.

Fercoq, Olivier and Richtárik, Peter. Smooth minimization of nonsmooth functions by parallel coordinate descent. *arXiv:1309.5885*, 2013.

Fercoq, Olivier, Qu, Zheng, Richtárik, Peter, and Takáč, Martin. Fast distributed coordinate descent for minimizing non-strongly convex losses. *IEEE International Workshop on Machine Learning for Signal Processing*, 2014.

Fountoulakis, Kimon and Tappenden, Rachael. Robust block coordinate descent. *arXiv:1407.7573*, 2014.

Glasmachers, Tobias and Dogan, Urun. Accelerated coordinate descent with adaptive coordinate frequencies. In *Asian Conference on Machine Learning*, pp. 72–86, 2013.

Jaggi, Martin, Smith, Virginia, Takac, Martin, Terhorst, Jonathan, Krishnan, Sanjay, Hofmann, Thomas, and Jordan, Michael I. Communication-efficient distributed dual coordinate ascent. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3068–3076. Curran Associates, Inc., 2014.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, 2013.

Konečný, Jakub and Richtárik, Peter. S2GD: Semi-stochastic gradient descent methods. *arXiv:1312.1666*, 2014.

Konečný, Jakub, Lu, Jie, Richtárik, Peter, and Takáč, Martin. mS2GD: Mini-batch semi-stochastic gradient descent in the proximal setting. *arXiv:1410.4744*, 2014a.

Konečný, Jakub, Qu, Zheng, and Richtárik, Peter. Semi-stochastic coordinate descent. *arXiv:1412.6293*, 2014b.

Lin, Qihang, Lu, Zhaosong, and Xiao, Lin. An accelerated proximal coordinate gradient method and its application to regularized empirical risk minimization. Technical Report MSR-TR-2014-94, July 2014.

Liu, Ji and Wright, Stephen J. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *arXiv:1403.3862*, 2014.

Loshchilov, I., Schoenauer, M., and Sebag, M. Adaptive Coordinate Descent. In et al., N. Krasnogor (ed.), *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 885–892. ACM Press, July 2011.

Lukasewitz, Isabella. Block-coordinate frank-wolfe optimization. A study on randomized sampling methods, 2013.

Mahajan, Dhruv, Keerthi, S. Sathiya, and Sundararajan, S. A distributed block coordinate descent method for training l1 regularized linear classifiers. *arXiv:1405.4544*, 2014.

Mairal, Julien. Incremental majorization-minimization optimization with application to large-scale machine learning. Technical report, 2014.

Mareček, Jakub, Richtárik, Peter, and Takáč, Martin. Distributed block coordinate descent for minimizing partially separable functions. *arXiv:1406.0328*, 2014.

Necoara, Ion and Patrascu, Andrei. A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. *Computational Optimization and Applications*, 57:307–337, 2014.

Necoara, Ion, Nesterov, Yurii, and Glineur, Francois. Efficiency of randomized coordinate descent methods on optimization problems with linearly coupled constraints. Technical report, Politehnica University of Bucharest, 2012.

Nemirovski, Arkadi, Juditsky, Anatoli, Lan, Guanghui, and Shapiro, Alexander. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

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

Qu, Zheng and Richtárik, Peter. Coordinate descent methods with arbitrary sampling I: Algorithms and complexity. *arXiv:1412.8060*, 2014.

Qu, Zheng and Richtárik, Peter. Coordinate Descent with Arbitrary Sampling II: Expected Separable Overapproximation. *ArXiv e-prints*, 2014.

Qu, Zheng, Richtárik, Peter, and Zhang, Tong. Randomized Dual Coordinate Ascent with Arbitrary Sampling. *arXiv:1411.5873*, 2014.

Richtárik, Peter and Takáč, Martin. Distributed coordinate descent method for learning with big data. *arXiv:1310.2059*, 2013a.

Richtárik, Peter and Takáč, Martin. On optimal probabilities in stochastic coordinate descent methods. *arXiv:1310.3438*, 2013b.

Richtárik, Peter and Takáč, Martin. Parallel coordinate descent methods for big data optimization problems. *Mathematical Programming (after minor revision), arXiv:1212.0873*, 2012.

Richtárik, Peter and Takáč, Martin. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(2):1–38, 2014.

Schaul, Tom, Zhang, Sixin, and LeCun, Yann. No more pesky learning rates. *Journal of Machine Learning Research*, 3(28):343–351, 2013.

Schmidt, Mark, Le Roux, Nicolas, and Bach, Francis. Minimizing finite sums with the stochastic average gradient. *arXiv:1309.2388*, 2013.

Shalev-Shwartz, Shai and Tewari, Ambuj. Stochastic methods for ℓ_1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, 2011.

Shalev-Shwartz, Shai and Zhang, Tong. Proximal stochastic dual coordinate ascent. *arXiv:1211.2717*, 2012.

Shalev-Shwartz, Shai and Zhang, Tong. Accelerated mini-batch stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems 26*, pp. 378–385. 2013a.

Shalev-Shwartz, Shai and Zhang, Tong. Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research*, 14(1):567–599, 2013b.

Takáč, Martin, Bijral, Avleen Singh, Richtárik, Peter, and Srebro, Nathan. Mini-batch primal and dual methods for svms. *CoRR*, abs/1303.2314, 2013.

Tappenden, Rachael, Richtárik, Peter, and Gondzio, Jacek. Inexact block coordinate descent method: complexity and preconditioning. *arXiv:1304.5530*, 2013.

Tappenden, Rachael, Richtárik, Peter, and Büke, Burak. Separable approximations and decomposition methods for the augmented lagrangian. *Optimization Methods and Software*, 2014.

Trofimov, Ilya and Genkin, Alexander. Distributed coordinate descent for l1-regularized logistic regression. *arXiv:1411.6520*, 2014.

Wright, Stephen J. Coordinate descent algorithms. Technical report, 2014. URL http://www.optimization-online.org/DB_FILE/2014/12/4679.pdf.

Xiao, Lin and Zhang, Tong. A proximal stochastic gradient method with progressive variance reduction. *arXiv:1403.4699*, 2014.

Zhang, Yuchen and Xiao, Lin. Stochastic primal-dual coordinate method for regularized empirical risk minimization. Technical Report MSR-TR-2014-123, September 2014.

Zhao, Peilin and Zhang, Tong. Stochastic optimization with importance sampling. *arXiv:1401.2753*, 2014.

Zhao, Tuo, Liu, Han, and Zhang, Tong. A general theory of pathwise coordinate optimization. *arXiv:1412.7477*, 2014a.

Zhao, Tuo, Yu, Mo, Wang, Yiming, Arora, Raman, and Liu, Han. Accelerated mini-batch randomized block coordinate descent method. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3329–3337. Curran Associates, Inc., 2014b.

Appendix

Proofs

We shall need the following inequality.

Lemma 13. *Function $f:\mathbb{R}^n\to\mathbb{R}$ defined in (3) satisfies the following inequality:*

$$f(\alpha+h)\leq f(\alpha)+\langle\nabla f(\alpha),h\rangle+\frac{1}{2\lambda n^2}h^\top A^\top Ah, \quad (18)$$

holds for $\forall\alpha,h\in\mathbb{R}^n$.

Proof. Since g is 1-strongly convex, g^* is 1-smooth. Pick $\alpha,h\in\mathbb{R}^n$. Since, $f(\alpha)=\lambda g^*(\frac{1}{\lambda n}A\alpha)$, we have

$$\begin{aligned}
&f(\alpha+h)=\lambda g^*(\frac{1}{\lambda n}A\alpha+\frac{1}{\lambda n}Ah)\\
&\leq\lambda\left(g^*(\frac{1}{\lambda n}A\alpha)+\langle\nabla g^*(\frac{1}{\lambda n}A\alpha),\frac{1}{\lambda n}Ah\rangle+\frac{1}{2}\|\frac{1}{\lambda n}Ah\|^2\right)\\
&=f(\alpha)+\langle\nabla f(\alpha),h\rangle+\frac{1}{2\lambda n^2}h^\top A^\top Ah.
\end{aligned}$$

□

Proof of Lemma 3. It can be easily checked that the following relations hold

$$\nabla_i f(\alpha^t)=\frac{1}{n}A_i^\top w^t,\ \ \forall t\geq 0,\ \ i\in[n], \quad (19)$$

$$g(w^t)+g^*(\bar\alpha^t)=\langle w^t,\bar\alpha^t\rangle,\ \ \forall t\geq 0, \quad (20)$$

where $\{w^t,\alpha^t,\bar\alpha^t\}_{t\geq 0}$ is the output sequence of Algorithm 1. Let $t\geq 0$ and $\theta\in[0,\min_i p_i^t]$. For each $i\in[n]$, since ϕ_i is $1/\gamma$-smooth, ϕ_i^* is γ-strongly convex and thus for arbitrary $s_i\in[0,1]$,

$$\begin{aligned}
&\phi_i^*(-\alpha_i^t+s_i\kappa_i^t)\\
&=\phi_i^*\left((1-s_i)(-\alpha_i^t)+s_i\nabla\phi_i(A_i^\top w^t)\right)\\
&\leq(1-s_i)\phi_i^*(-\alpha_i^t)+s_i\phi_i^*(\nabla\phi_i(A_i^\top w^t))\\
&\quad-\frac{\gamma s_i(1-s_i)|\kappa_i^t|^2}{2}.
\end{aligned} \quad (21)$$

We have:

$$\begin{aligned}
&f(\alpha^{t+1})-f(\alpha^t)\\
&\overset{(18)}{\leq}\langle\nabla f(\alpha^t),\alpha^{t+1}-\alpha^t\rangle\\
&\quad+\frac{1}{2\lambda n^2}\langle\alpha^{t+1}-\alpha^t,A^\top A(\alpha^{t+1}-\alpha^t)\rangle\\
&=\nabla_i f(\alpha^t)\Delta\alpha_{i_t}^t+\frac{v_i}{2\lambda n^2}|\Delta\alpha_{i_t}^t|^2\\
&\overset{(19)}{=}\frac{1}{n}A_{i_t}^\top w^t\Delta\alpha_{i_t}^t+\frac{v_i}{2\lambda n^2}|\Delta\alpha_{i_t}^t|^2
\end{aligned} \quad (22)$$

Thus,

$$\begin{aligned}
&D(\alpha^{t+1})-D(\alpha^t)\\
&\overset{(22)}{\geq}-\frac{1}{n}A_{i_t}^\top w^t\Delta\alpha_{i_t}^t-\frac{v_{i_t}}{2\lambda n^2}|\Delta\alpha_{i_t}^t|^2+\frac{1}{n}\sum_{i=1}^n\phi_i^*(-\alpha_i^t)\\
&\qquad-\frac{1}{n}\sum_{i=1}^n\phi_i^*(-\alpha_i^{t+1})\\
&=-\frac{1}{n}A_{i_t}^\top w^t\Delta\alpha_{i_t}^t-\frac{v_{i_t}}{2\lambda n^2}|\Delta\alpha_{i_t}^t|^2+\frac{1}{n}\phi_{i_t}^*(-\alpha_{i_t}^t)\\
&\quad-\frac{1}{n}\phi_{i_t}^*(-\left(\alpha_{i_t}^t+\Delta\alpha_{i_t}^t\right))\\
&=\max_{\Delta\in\mathbb{R}}-\frac{1}{n}A_{i_t}^\top w^t\Delta-\frac{v_{i_t}}{2\lambda n^2}|\Delta|^2+\frac{1}{n}\phi_{i_t}^*(-\alpha_{i_t}^t)\\
&\quad-\frac{1}{n}\phi_{i_t}^*(-\left(\alpha_{i_t}^t+\Delta\right)),
\end{aligned}$$

where the last equality follows from the definition of $\Delta\alpha_{i_t}^t$ in Algorithm 1. Then by letting $\Delta=-s_i\kappa_{i_t}^t$ for some arbitrary $s_i\in[0,1]$ we get:

$$\begin{aligned}
&D(\alpha^{t+1})-D(\alpha^t)\\
&\geq\frac{s_iA_{i_t}^\top w^t\kappa_{i_t}^t}{n}-\frac{s_i^2v_{i_t}|\kappa_{i_t}^t|^2}{2\lambda n^2}+\frac{1}{n}\phi_{i_t}^*(-\alpha_{i_t}^t)\\
&\quad-\frac{1}{n}\phi_{i_t}^*(-\alpha_{i_t}^t+s_i\kappa_{i_t}^t)\\
&\overset{(21)}{\geq}\frac{s_i}{n}\left(\phi_{i_t}^*(-\alpha_{i_t}^t)-\phi_{i_t}^*(\nabla\phi_{i_t}(A_{i_t}^\top w^t))+A_{i_t}^\top w^t\kappa_{i_t}^t\right)\\
&\quad-\frac{s_i^2v_{i_t}|\kappa_{i_t}^t|^2}{2\lambda n^2}+\frac{\gamma s_i(1-s_i)|\kappa_{i_t}^t|^2}{2n}.
\end{aligned}$$

By taking expectation with respect to i_t we get:

$$\begin{aligned}
&\mathbb{E}_t\left[D(\alpha^{t+1})-D(\alpha^t)\right]\\
&\geq\sum_{i=1}^n\frac{p_i^ts_i}{n}\left[\phi_i^*(-\alpha_i^t)-\phi_i^*(\nabla\phi_i(A_i^\top w^t))+A_i^\top w^t\kappa_i^t\right]\\
&\quad-\sum_{i=1}^n\frac{p_i^ts_i^2|\kappa_i^t|^2(v_i+\lambda\gamma n)}{2\lambda n^2}+\sum_{i=1}^n\frac{p_i^t\gamma s_i|\kappa_i^t|^2}{2n}.
\end{aligned} \quad (23)$$

Set

$$s_i=\begin{cases}0, & i\notin I_t\\ \theta/p_i^t, & i\in I_t\end{cases} \quad (24)$$

Then $s_i\in[0,1]$ for each $i\in[n]$ and by plugging it into (23) we get:

$$\begin{aligned}
&\mathbb{E}_t\left[D(\alpha^{t+1})-D(\alpha^t)\right]\\
&\geq\frac{\theta}{n}\sum_{i\in I_t}\left[\phi_i^*(-\alpha_i^t)-\phi_i^*(\nabla\phi_i(A_i^\top w^t))+A_i^\top w^t\kappa_i^t\right]\\
&\quad-\frac{\theta}{2\lambda n^2}\sum_{i\in I_t}\left(\frac{\theta(v_i+n\lambda\gamma)}{p_i^t}-n\lambda\gamma\right)|\kappa_i^t|^2
\end{aligned}$$

Finally note that:

$$
\begin{aligned}
&P(w^t) - D(\alpha^t) \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[\phi_i(A_i^\top w^t) + \phi_i^*(-\alpha_i^t)\right] + \lambda\left(g(w^t) + g^*(\bar{\alpha}^t)\right) \\
&\overset{(20)}{=} \frac{1}{n}\sum_{i=1}^{n}\left[\phi_i^*(-\alpha_i^t) + \phi_i(A_i^\top w^t)\right] + \frac{1}{n}\langle w^t, A\alpha^t\rangle \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[\phi_i^*(-\alpha_i^t) + A_i^\top w^t \nabla\phi_i(A_i^\top w^t)\right. \\
&\qquad\qquad \left. -\phi_i^*(\nabla\phi_i(A_i^\top w^t)) + A_i^\top w^t \alpha_i^t\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[\phi_i^*(-\alpha_i^t) - \phi_i^*(\nabla\phi_i(A_i^\top w^t)) + A_i^\top w^t \kappa_i^t\right] \\
&= \frac{1}{n}\sum_{i\in I_t}\left[\phi_i^*(-\alpha_i^t) - \phi_i^*(\nabla\phi_i(A_i^\top w^t)) + A_i^\top w^t \kappa_i^t\right]
\end{aligned}
$$

□

Proof of Lemma 4. Note that (13) is a standard constrained maximization problem, where everything independent of p can be treated as a constant. We define the Lagrangian

$$L(p,\eta) = \theta(\kappa, p) - \eta(\sum_{i=1}^{n} p_i - 1)$$

and get the following optimality conditions:

$$
\begin{aligned}
&\frac{|\kappa_i^t|^2(v_i + n\lambda\gamma)}{p_i^2} = \frac{|\kappa_j^t|^2(v_j + n\lambda\gamma)}{p_j^2},\ \forall i,j \in [n] \\
&\sum_{i=1}^{n} p_i = 1 \\
&p_i \geq 0,\ \ \forall i \in [n],
\end{aligned}
$$

the solution of which is (14). □

Proof of Lemma 5. Note that in the proof of Lemma 3, the condition $\theta \in [0, \min_{i\in I_t} p_i^t]$ is only needed to ensure that s_i defined by (24) is in $[0,1]$ so that (21) holds. If ϕ_i is quadratic function, then (21) holds for arbitrary $s_i \in \mathbb{R}$. Therefore in this case we only need θ to be positive and the same reasoning holds. □