SDNA: Stochastic Dual Newton Ascent for Empirical Risk Minimization

Zheng Qu ZHENGQU@HKU.HK
Department of Mathematics, The University of Hong Kong, Hong Kong

Peter Richtárik PETER.RICHTARIK@ED.AC.UK
School of Mathematics, The University of Edinburgh, UK

Martin Takáč TAKAC.MT@GMAIL.COM
Industrial and Systems Engineering, Lehigh University, USA

Olivier Fercoq OLIVIER.FERCOQ@TELECOM-PARISTECH.FR
LTCI, CNRS, Télécom Paris-Tech, Université Paris-Saclay, France

Abstract

We propose a new algorithm for minimizing regularized empirical loss: *Stochastic Dual Newton Ascent (SDNA)*. Our method is dual in nature: in each iteration we update a random subset of the dual variables. However, unlike existing methods such as stochastic dual coordinate ascent, SDNA is capable of utilizing *all local curvature information* contained in the examples, which leads to striking improvements in both theory and practice – sometimes by orders of magnitude. In the special case when an L2-regularizer is used in the primal, the dual problem is a concave quadratic maximization problem plus a separable term. In this regime, SDNA in each step solves a proximal subproblem involving a random principal submatrix of the Hessian of the quadratic function; whence the name of the method.

1. Introduction

Empirical risk minimization (ERM) is a fundamental paradigm in the theory and practice of statistical inference and machine learning (Shalev-Shwartz & Ben-David, 2014). In the "big data" era it is increasingly common in practice to solve ERM problems with a massive number of examples, which leads to new algorithmic challenges. State-of-the-art optimization methods for ERM include i) stochastic (sub)gradient descent (Shalev-Shwartz et al., 2011; Takáč et al., 2013), ii) methods based on stochastic estimates of the gradient with diminishing variance such as SAG (Schmidt et al., 2013), SVRG (Johnson & Zhang, 2013), S2GD (Konečný & Richtárik, 2014), proxSVRG (Xiao & Zhang, 2014), MISO (Mairal, 2015), SAGA (Defazio et al., 2014), minibatch S2GD (Konečný et al., 2014a), S2CD (Konečný et al., 2014b), and iii) variants of stochastic dual coordinate ascent (Shalev-Shwartz & Zhang, 2013b; Zhao & Zhang, 2014; Takáč et al., 2013; Shalev-Shwartz & Zhang, 2013a;a; Lin et al., 2014; Qu et al., 2014).

There have been several attempts at designing methods that combine randomization with the use of curvature (second-order) information. For example, methods based on running coordinate ascent in the dual such as those mentioned above and also (Richtárik & Takáč, 2014; 2015; Fercoq & Richtárik, 2013; Tappenden et al., 2014; Richtárik & Takáč, 2013; 2015; Fercoq & Richtárik, 2015; Fercoq et al., 2014; Qu et al., 2014; Qu & Richtárik, 2014a) use curvature information contained in the diagonal of a bound on the Hessian matrix. Block coordinate descent methods, when equipped with suitable data-dependent norms for the blocks, use information contained in the block diagonal of the Hessian (Tappenden et al., 2013). A more direct route to incorporating curvature information was taken by (Schraudolph et al., 2007) in their stochastic L-BFGS method and by (Byrd et al., 2014) and (Sohl-Dickstein et al., 2014) in their stochastic quasi-Newton methods. Complexity estimates are not easy to find. An exception in this regard is the work of (Bordes et al., 2009), who give a $O(1/\epsilon)$ complexity bound for a Quasi-Newton SGD method.

An alternative approach is to consider block coordinate descent methods with overlapping blocks (Tseng & Yun, 2009; Fountoulakis & Tappenden, 2015). While typically efficient in practice, none of the methods mentioned above are equipped with complexity bounds (bounds on the number of iterations). Tseng and Yun (Tseng & Yun, 2009) only showed convergence to a stationary point and focus on non-overlapping blocks for the rest of their paper. Numerical evidence that this approach is promising is provided

Proceedings of the 33^{rd} *International Conference on Machine Learning*, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

in (Fountoulakis & Tappenden, 2015) with some mild convergence rate results but no iteration complexity.

The main contribution of this paper is the analysis of stochastic block coordinate descent methods with overlapping blocks. We then instantiate this to get a new algorithm—**Stochastic Dual Newton Ascent (SDNA)**—for solving a regularized ERM problem with smooth loss functions and a strongly convex regularizer (primal problem). Our method is stochastic in nature and has the capacity to utilize *all curvature information* inherent in the data. While we do our analysis for an arbitrary strongly convex regularizer, for the purposes of the introduction we shall describe the method in the case of the L2 regularizer. In this case, the dual problem is a concave quadratic maximization problem with a strongly concave separable penalty.

SDNA in each iteration picks a random subset of the dual variables (which corresponds to picking a minibatch of examples in the primal problem), following an arbitrary probability law, and maximizes, exactly, the dual objective restricted to the random subspace spanned by the coordinates. Equivalently, this can be seen as the solution of a proximal subproblem involving a random principal submatrix of the Hessian of the quadratic function. Hence, SDNA utilizes all curvature information available in the random subspace in which it operates. Note that this is very different from the update strategy of parallel / minibatch coordinate descent methods. Indeed, while these methods also update a random subset of variables in each iteration, they instead only utilize curvature information present in the diagonal of the Hessian.

In the case of quadratic loss, and when viewed as a primal method, SDNA can be interpreted as a variant of the recently introduced Iterative Hessian Sketch algorithm (Pilanci & Wainwright, 2014).

SDCA-like methods need *more* passes through data to converge as the minibatch size increases. However, SDNA enjoys the opposite behavior: with increasing minibatch size, up to a certain threshold , SDNA needs fewer passes through data to converge. This observation is confirmed by our numerical experiments.

In particular, we show that the expected duality gap decreases at a geometric rate which i) is better than that of SDCA-like methods such as SDCA (Shalev-Shwartz & Zhang, 2013b) and QUARTZ (Qu et al., 2014), and ii) improves with increasing minibatch size. This improvement does not come for free: as we increase the minibatch size, the subproblems grow in size as they involve larger portions of the Hessian. We find through experiments that for some, especially dense problems, even relatively small minibatch sizes lead to dramatic speedups in actual runtime.

En route to developing SDNA which we describe in Section 5, we also analyze several other algorithms: two in Section 2 where we focus on smooth problems and a novel minibatch variant of SDCA in Section 5, for the sake of finding suitable method to compare SDNA to. SDNA is equivalent to applying the proximal variant of the method developed in Section 2 to the dual of the ERM problem. However, as we are mainly interested in solving the ERM (primal) problem, we additionally prove that the expected duality gap decreases at a geometric rate. Our technique for doing this is a variant of the one use by (Shalev-Shwartz & Zhang, 2013b), but generalized to an arbitrary sampling.

Notation. In the paper we use the following notation. By $e_1, \dots, e_n$ we denote the standard basis vectors in $\mathbb{R}^n$. For any $x \in \mathbb{R}^n$, we denote by x_i the ith element of x, i.e., $x_i = e_i^\top x$. For any two vectors x, y of equal size, we write $\langle x, y \rangle = x^\top y = \sum_i x_i y_i$. $\mathbf{I}$ is the identity matrix in $\mathbb{R}^{n\times n}$ and $\mathbf{D}(w)$ is the diagonal matrix in $\mathbb{R}^{n\times n}$ with $w \in \mathbb{R}^n$ on its diagonal. We will write $\mathbf{M} \succeq 0$ (resp. $\mathbf{M} \succ 0$) to indicate that $\mathbf{M}$ is positive semidefinite (resp. positive definite). Let S be a nonempty subset of $[n] := \{1, 2, \dots, n\}$. For any matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ we write $\mathbf{M}_S$ for the matrix obtained from $\mathbf{M}$ by retaining elements $\mathbf{M}_{ij}$ for which both $i \in S$ and $j \in S$ and zeroing out all other elements. For any vector $h \in \mathbb{R}^n$ we write h_S for the vector obtained by retaining elements h_i with $i \in S$ and zeroing out the others, i.e., $h_S := \mathbf{I}_S h = \sum_{i\in S} h_i e_i$. By $(\mathbf{M}_S)^{-1}$ we denote the matrix $\mathbf{Z}$ in $\mathbb{R}^{n\times n}$ for which

$$\mathbf{Z}\mathbf{M}_S = \mathbf{M}_S\mathbf{Z} = \mathbf{I}_S \qquad \text{and} \qquad \mathbf{Z}_S = \mathbf{Z}. \tag{1}$$

That is, $\mathbf{Z}$ is the $n \times n$ matrix containing the inverse of the $|S| \times |S|$ submatrix of $\mathbf{M}$ corresponding to elements $(i, j) \in S$ in the same position, while having all other elements equal to zero.

2. Minimization of a Smooth Function

We start by considering the following unconstrained minimization problem:

$$\min_{x\in\mathbb{R}^n} f(x). \tag{2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex differentiable function. In addition, we assume that there are two matrices $\mathbf{M}, \mathbf{G} \in \mathbb{R}^{n\times n}$ such that for all $x, h \in \mathbb{R}^n$,

$$f(x) + \langle \nabla f(x), h\rangle + \tfrac{1}{2}\langle \mathbf{G}h, h\rangle \le f(x + h), \tag{3}$$

$$f(x + h) \le f(x) + \langle \nabla f(x), h\rangle + \tfrac{1}{2}\langle \mathbf{M}h, h\rangle. \tag{4}$$

Moreover, we assume that $\mathbf{G} \succ 0$. When $\mathbf{M} = L\mathbf{H}$ and $\mathbf{G} = \mu\mathbf{H}$ for some $L \ge \mu > 0$ and positive definite matrix $\mathbf{H}$, (3) and (4) mean that f is L-smooth and μ-strongly convex with respect to the norm $\|h\|_{\mathbf{H}} = \langle \mathbf{H}h, h\rangle^{1/2}$. Usually one simply assumes that $\mathbf{H} = \mathbf{I}$. However, in many situations we can get *more information* in $\mathbf{M}$ and $\mathbf{G}$, which we will utilize in this paper.

2.1. Three stochastic algorithms

We now describe three algorithmic strategies for solving problem (2). All these methods have the form

$$x^{k+1} \leftarrow x^k + h^k, \tag{5}$$

where h_i^k is only allowed to be nonzero for $i \in S_k$, where $\{S_k\}_{k\geq 0}$ are i.i.d. random subsets of $[n] := \{1, 2, \dots, n\}$ ("samplings"). That is, all methods in each iteration update a random subset of the variables and only differ in how the update elements h_i^k for $i \in S_k$ are computed. As $\{S_k\}_{k\geq 0}$ are i.i.d., it will be convenient to write $\hat{S}$ for a set-valued random variable which shares their distribution. For the methods to work, we need to require that every coordinate has a positive probability of being sampled. For technical reasons that will be apparent later, we will also assume that $\hat{S}$ is nonempty with probability 1. Thus the following conditions are assumed to hold throughout the paper:

$$p_i := \mathbb{P}(i \in \hat{S}) > 0,\ i \in [n] \quad \text{and} \quad \mathbb{P}(\hat{S} = \emptyset) = 0. \tag{6}$$

Sampling satisfying the first (resp. second) condition will be referred to as a proper (resp. nonvacuous) sampling.

Method 1 (overlapping-block coordinate descent). In the first method, we compute $(\mathbf{M}_{S_k})^{-1}$ and set

$$h^k = -(\mathbf{M}_{S_k})^{-1} \nabla f(x^k). \tag{Method 1}$$

Note that $(\mathbf{M}_{S_k})^{-1}$ is well defined if $\mathbf{M} \succ 0$, which we assume throughout. The computation of h^k involves the solution of a linear system involving an $|S_k| \times |S_k|$ matrix. Equivalently (in principle, not in terms of computational effort), the method involves inversion of a random principal submatrix of $\mathbf{M}$ of size $|S_k| \times |S_k|$. Also note that we only need to compute elements $i \in S_k$ of the gradient $\nabla f(x_k)$, since in view of (1), $(\mathbf{M}_{S_k})^{-1}$ only acts on those elements. If $|S_k|$ is reasonably small, the computation of h^k is cheap.

Method 2. We compute the inverse[1] of $\mathbb{E}[\mathbf{M}_{\hat{S}}]$ and set

$$h^k = -\mathbf{I}_{S_k} (\mathbb{E}[\mathbf{M}_{\hat{S}}])^{-1} \mathbf{D}(p) \nabla f(x^k). \tag{Method 2}$$

This strategy is easily implementable when $|\hat{S}| = 1$ with probability 1 (i.e., if we update a single variable only). This is because then $\mathbb{E}[\mathbf{M}_{\hat{S}}]$ is a diagonal matrix with the (i, i) element equal to $p_i \mathbf{M}_{ii}$. For more complicated samplings $\hat{S}$, however, the matrix $\mathbb{E}[\mathbf{M}_{\hat{S}}]$ will be as hard to invert as $\mathbf{M}$. Hence, in many situations, Method 2 is impractical. However, we include it in the discussion for the sake of comparison – it will help us better understand the relationship between Method 1 (which contains a key new idea of this paper) and Method 3, which we shall describe next.

In (Qu & Richtárik, 2014a) it was shown that

$$\mathbb{E}\left[\mathbf{M}_{\hat{S}}\right] = \mathbf{P} \circ \mathbf{M}, \tag{7}$$

where $\circ$ denotes the Hadamard (element-wise) product of two matrices, and $\mathbf{P}$ is the $n \times n$ matrix with entries $\mathbf{P}_{ij} = \mathbb{P}(\{i, j\} \subseteq \hat{S})$. It can be easily shown that $\mathbf{P}$ is positive semidefinite.

Method 3 (parallel coordinate descent). Here we compute a vector $v \in \mathbb{R}^n$ for which

$$\mathbb{E}[\mathbf{M}_{\hat{S}}] \preceq \mathbf{D}(p)\mathbf{D}(v) \tag{8}$$

and then set

$$h^k = -\mathbf{I}_{S_k} (\mathbf{D}(v))^{-1} \nabla f(x^k). \tag{Method 3}$$

It can be shown that safe (albeit conservative) choice of v satisfying (8) is $v_i = \tau$, where τ is a number satisfying $\mathbb{P}(|\hat{S}| \leq \tau) = 1$. This and tighter bounds can be found in (Qu & Richtárik, 2014b). Hence, the update is clearly very easy to perform, and can be equivalently written as

$$h_i^k = \begin{cases} -\frac{1}{v_i} \langle e_i, \nabla f(x^k) \rangle, & i \in S_k \\ 0, & i \notin S_k. \end{cases} \tag{9}$$

We see that this method takes a coordinate descent step for every coordinate $i \in S_k$, with stepsize $1/v_i$. For a calculus allowing the computation of closed form formulas for v as a function of the sampling $\hat{S}$ we refer the reader to (Qu & Richtárik, 2014b). Method 3 was proposed and analyzed in (Richtárik & Takáč, 2015) as is known as "NSync". If the coordinates $i \in S_k$ of the gradient $\nabla f(x^k)$ are available, the updates h_i^k can be computed independently of each other. In particular, they can be trivially computed in parallel. For this reason, this method can be thought of as a parallel/minibatch coordinate descent method (Richtárik & Takáč, 2015). In fact, it is the first such method which was analyzed for an arbitrary sampling of coordinates.

Remark 1. It is easy to see that all three methods coincide if $|\hat{S}| = 1$ with probability 1. Moreover, Methods 1 and 2 coincide if $\hat{S} = [n]$ with probability 1.

2.2. Three linear convergence rates

We shall now show that, putting the issue of the cost of each iteration of the three methods aside, all enjoy a linear rate of convergence.

Theorem 2. *Let (3), (4) and (6) hold with $\mathbf{G} \succ 0$. Let $\{x^k\}_{k\geq 0}$ be the sequence of random vectors produced by Method m, for $m = 1, 2, 3$ and let x^* be the optimal solution of (2). Then*

$$\mathbb{E}[f(x^{k+1}) - f(x^*)] \leq (1 - \sigma_m)\, \mathbb{E}[f(x^k) - f(x^*)],$$

[1] The matrix $\mathbb{E}[\mathbf{M}_{\hat{S}}]$ is invertible if (6) holds and $\mathbf{M} \succ 0$, for a proof see the supplementary material.

where

$$\sigma_1 := \lambda_{\min}\left(\mathbf{G}^{1/2}\,\mathbb{E}\left[\left(\mathbf{M}_{\hat{S}}\right)^{-1}\right]\mathbf{G}^{1/2}\right), \qquad (10)$$

$$\sigma_2 := \lambda_{\min}\left(\mathbf{G}^{1/2}\mathbf{D}(p)\left(\mathbb{E}\left[\mathbf{M}_{\hat{S}}\right]\right)^{-1}\mathbf{D}(p)\mathbf{G}^{1/2}\right), \quad (11)$$

$$\sigma_3 := \lambda_{\min}\left(\mathbf{G}^{1/2}\mathbf{D}(p)\mathbf{D}(v^{-1})\mathbf{G}^{1/2}\right). \qquad (12)$$

That is,

$$k \geq \tfrac{1}{\sigma_m}\log\left(\tfrac{f(x^0)-f(x^*)}{\epsilon}\right) \Rightarrow \mathbb{E}[f(x^k)-f(x^*)] \leq \epsilon. \quad (13)$$

We will show in the next section that Method 1 has the fastest rate, followed by Method 2 and finally, Method 3.

3. Three Complexity Rates: Relationships and Properties

In this section we give various insights into the quantities σ_1, σ_2 and σ_3. First, in Section 3.1 we establish that $\sigma_1 \geq \sigma_2 \geq \sigma_3$, and illustrate the possible difference between these quantities on a simple example in three dimensions. In Section 3.2 we study the dependence of σ_2 on the sampling $\hat{S}$, and show how the complexity of Method 2, which always upper bounds the complexity of Method 1, improves as the size of $\hat{S}$ grows.

3.1. Ordering the rates

We now establish an important relationship between the quantities σ_1, σ_2 and σ_3, which sheds light on the convergence rates of the three methods.

Lemma 3. *If $\mathbf{M} \succeq 0$, then for any sampling we have $\mathbb{E}\left[\mathbf{M}_{\hat{S}}\right] \succeq 0$. If, moreover, $\mathbf{M} \succ 0$, and $\hat{S}$ is a proper sampling, then $\mathbb{E}\left[\mathbf{M}_{\hat{S}}\right] \succ 0$.*

Lemma 4. *If $\mathbf{M} \succ 0$ and $\hat{S}$ is a proper and nonvacuous sampling, then*

$$0 \prec \mathbf{D}(p)\left(\mathbb{E}\left[\mathbf{M}_{\hat{S}}\right]\right)^{-1}\mathbf{D}(p) \preceq \mathbb{E}\left[\left(\mathbf{M}_{\hat{S}}\right)^{-1}\right]. \qquad (14)$$

Lemma 5. *Assume $\mathbf{M} \succ 0$ and let $S \subseteq [n]$ be nonempty. Then*

$$(\mathbf{M}_S)^{-1} \preceq (\mathbf{M}^{-1})_S. \qquad (15)$$

We can now state and prove the main theorem.

Theorem 6. *Under the assumptions of Theorem 2, the quantities σ_1, σ_2 and σ_3 satisfy the following relations:*

$$0 < \sigma_3 \leq \sigma_2 \leq \sigma_1 \leq \min_{1 \leq i \leq n} p_i.$$

While the above result says that in terms of iteration complexity, Method 1 is better than Method 2, which in turn is better than Method 3, it does not quantify the difference. We now use a simple example with a quadratic function in 3 dimensions to illustrate that σ_1 can indeed be massively larger than σ_2 and σ_3.

Example 7 (A quadratic in 3D)**.** *Consider the function $f : \mathbb{R}^3 \to \mathbb{R}$ given by $f(x) = \frac{1}{2}x^T\mathbf{M}x$, with*

$$\mathbf{M} = \begin{pmatrix} 1.0000 & 0.9900 & 0.9999 \\ 0.9900 & 1.0000 & 0.9900 \\ 0.9999 & 0.9900 & 1.0000 \end{pmatrix}.$$

Note that Assumption (4) holds, and Assumption (3) holds with $\mathbf{G} = \mathbf{M}$. Let $\hat{S}$ be the "2-nice sampling" on $[n] = \{1, 2, 3\}$. That is, we set $\mathbb{P}(\hat{S} = \{i, j\}) = \frac{1}{3}$ for $(i, j) \in \{(1, 2), (2, 3), (3, 1)\}$. It can be verified that (8) holds with $v = (2, 2, 2)$; see (Richtárik & Takáč, 2015) or (Qu & Richtárik, 2014b). Therefore, $\mathbf{D}(p)\mathbf{D}(v^{-1}) = \frac{1}{3}\mathbf{I}$ and from a straightforward calculation we obtain:

$$\sigma_1 \approx 0.3350, \quad \sigma_2 \approx 1.333 \times 10^{-4}, \quad \sigma_3 \approx 0.333 \times 10^{-4}.$$

Note that in this case, the theoretical rate, σ_1, of Method 1 is 10,000 times better than the rate, σ_3, of parallel coordinate descent (Method 3).

This analysis is the first to prove such a good rate of convergence for a coordinate-descent type method on such an ill-conditioned problem. For instance, the algorithm of (Fountoulakis & Tappenden, 2015) has a rate $\sigma_{\text{FT}} \leq 1.995 \times 10^{-4}$.

3.2. Dependence of σ_2 on the sampling

As mentioned above, in this section we shall study the dependence of σ_2 on the sampling $\hat{S}$. To this goal, we shall consider a parametric family of samplings described by a single parameter, τ, for which it is easy to formalize the notion of "growth", as this coincides with the growth of the parameter τ itself.

In particular, we consider the family of τ-nice samplings (Richtárik & Takáč, 2015), for $\tau \in [n]$. Informally, a sampling $\hat{S}$ is called τ-nice, if it only picks subsets of $[n]$ of cardinality τ, uniformly at random. More formally, a τ-nice sampling is defined by the probability mass function as follows: $\mathbb{P}(\hat{S} = S) = 1/\binom{n}{\tau}$ for all $S \subseteq [n]$ for which $|S| = \tau$.

Further, if $\hat{S}$ is the τ-nice sampling, define[2]

$$\mathbf{C}_\tau \overset{\text{def}}{=} \mathbf{D}(p)^{-1}\,\mathbb{E}\left[\mathbf{M}_{\hat{S}}\right]\mathbf{D}(p)^{-1}. \qquad (16)$$

This is the inverse of one of the quantities appearing in (14). We have the following result.

Lemma 8. *Assume that $\mathbf{M} \succeq 0$. Then:*

(i) The mapping $\tau \mapsto \mathbf{C}_\tau$ is monotone decreasing. That is, if $1 \leq \tau_2 \leq \tau_1 \leq n$, then $\mathbf{C}_{\tau_1} \preceq \mathbf{C}_{\tau_2}$.

(ii) If, moreover, $\mathbf{M} \succ 0$, then the mapping $\tau \mapsto \mathbf{C}_\tau^{-1}$ is monotone increasing.

[2] Recall that $p = (p_1, \ldots, p_n)$, where $p_i = \mathbb{P}(i \in \hat{S})$, and that $\mathbf{D}(p)$ is the diagonal matrix with vector p on the diagonal.

If $\mathbf{M}$ is positive definite (as in Assumption (4)), we have the following result.

Theorem 9. *Let Assumptions* (3) *and* (4) *hold. Further, let $\hat{S}$ be the τ-nice sampling for $\tau \in [n]$ and let $\sigma_2 = \sigma_2(\tau)$ be the rate of Method 2, as described in* (11)*. Then*

$$\frac{1}{\sigma_2(\tau)} = \tfrac{n}{n-1}\lambda_{\max}\left(\mathbf{G}^{-1/2}\left[\left(\tfrac{n}{\tau}-1\right)\mathbf{D}_{\mathbf{M}} + \left(1-\tfrac{1}{\tau}\right)\mathbf{M}\right]\mathbf{G}^{-1/2}\right), \tag{17}$$

and moreover, $\sigma_2(\tau)$ is monotonically increasing in τ.

Proof. In view of the definition of σ_2 in (11) and the definition of $\mathbf{C}_\tau$ in (16), we can write $\sigma_2(\tau) = \lambda_{\min}\left(\mathbf{G}^{1/2}\mathbf{C}_\tau^{-1}\mathbf{G}^{1/2}\right) = 1/\lambda_{\max}(\mathbf{G}^{-1/2}\mathbf{C}_\tau\mathbf{G}^{-1/2})$. The claim follows by plugging in for $\mathbf{C}_\tau$ from (16) and using Lemma 8. □

The above theorem should be interpreted as follows: As we increase the sampling size τ, the complexity of Method 2 improves. Since in view of Theorem 6 we have $\sigma_1(\tau) \geq \sigma_2(\tau)$, we should also expect the number of iterations of Method 1 to decrease as τ increases. Indeed, we observe this behavior also in our numerical experiments.

For more insight, consider now the special case when f is a quadratic so that $\mathbf{G} = \mathbf{M}$. From (17) we then get:

$$\frac{1}{\sigma_2(\tau)} = \frac{n}{n-1}\left(1 - \frac{1}{\tau} + \left(\frac{n}{\tau} - 1\right)\delta_f\right), \tag{18}$$

where $\delta_f \overset{\text{def}}{=} \lambda_{\max}\left(\mathbf{M}^{-1/2}\mathbf{D}_{\mathbf{M}}\mathbf{M}^{-1/2}\right) \geq 1$. By inspecting the expression in (18) as a function of τ, we notice that

$$\frac{1}{\sigma_2(\tau)} \leq \frac{1}{\sigma_2(1)\tau}$$

for all $\tau \geq 1$ (this is also true if in the case when $\mathbf{G} \neq \mathbf{M}$). That is, for instance, with $\tau = 100$, Method 1 will need *fewer* than one hundredth of the number of iterations to converge than for $\tau = 1$. We hence obtain the following result:

Corollary 10. *Method 2 (and hence also Method 1) enjoys* superlinear speedup *in τ.*

This phenomenon does not occur in parallel coordinate descent methods such as Method 3, where the most one can hope for is linear speedup.

4. Minimization of a Composite Function

In this section we consider the following *composite* minimization problem:

$$\min_{x\in\mathbb{R}^n} F(x) \equiv f(x) + \sum_{i=1}^{n} \psi_i(x_i). \tag{19}$$

We assume that f satisfies Assumptions (3) and (4). The difference from the setup in the previous section is in the inclusion of the separable term $\sum_i \psi_i$. In addition, we assume that for each i, $\psi_i : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is closed and γ_i-strongly convex for some $\gamma_i \geq 0$. We shall write $\gamma = (\gamma_1, \dots, \gamma_n) \in \mathbb{R}^n_+$.

4.1. Proximal overlapping-block coordinate descent

We now propose Algorithm 1, which is a variant of Method 1 applicable to problem (19). If $\psi_i \equiv 0$ for all i, the methods coincide.

This algorithm is closely related to Flexible Block Coordinate Descent in (Fountoulakis & Tappenden, 2015). They mainly differ by the fact that we concentrate on a fixed global surrogate of the Hessian matrix while (Fountoulakis & Tappenden, 2015) allows a line search and local surrogates. On the other hand, we get more precise convergence results, as stated in Theorem 11: the method converges at a geometric rate, in expectation.

Algorithm 1 Proximal Overlapping-Block CD

```
Parameters: proper nonvacuous sampling $\hat{S}$
Initialization: choose initial point $x^0 \in \mathbb{R}^n$
for $k = 0, 1, 2, \dots$ do
    Generate a random set of blocks $S_k \sim \hat{S}$
    Compute: $h^k = \arg\min_{h\in\mathbb{R}^n} \langle \nabla f(x^k), h_{S_k}\rangle + \frac{1}{2}\langle h, \mathbf{M}_{S_k} h\rangle + \sum_{i\in S_k} \psi_i(x_i^k + h_i)$
    Update: $x^{k+1} := x^k + h^k_{S_k}$
end for
```

Theorem 11. *The output sequence $\{x^k\}_{k\geq 0}$ of Algorithm 1 satisfies:*

$$\mathbb{E}[F(x^{k+1}) - F(x^*)] \leq (1 - \sigma_1^{prox})\,\mathbb{E}[F(x^k) - F(x^*)],$$

where x^* is the solution of (19) *and σ_1^{prox} is given by*

$$\lambda_{\min}\Big[\mathbf{D}(p)\left(\mathbb{E}[\mathbf{M}_{\hat{S}}] + \mathbf{D}(p)\mathbf{D}(\gamma)\right)^{-1}\mathbf{D}(p)(\mathbf{D}(\gamma) + \mathbf{G})\Big].$$

Note that for positive definite matrices $\mathbf{X}, \mathbf{Y}$, we have

$$\lambda_{\min}(\mathbf{X}^{-1}\mathbf{Y}) = \lambda_{\min}(\mathbf{Y}^{1/2}\mathbf{X}^{-1}\mathbf{Y}^{1/2}).$$

It is this latter form we have used in the formulation of Theorem 2. If $\gamma \equiv 0$ (ψ_i are merely convex), we have

$$\sigma_1^{prox} = \lambda_{\min}(\mathbf{G}^{1/2}\mathbf{D}(p)(\mathbb{E}[\mathbf{M}_{\hat{S}}])^{-1}\mathbf{D}(p)\mathbf{G}^{1/2}).$$

Note that while this rate applies to a proximal/composite variant of Method 1, its rate is best compared to the rate σ_2 of Method 2. Indeed, from (11) and Theorem 6, we get

$$\sigma_1 \geq \sigma_2 = \sigma_1^{prox}.$$

So, the rate we can prove for the composite version of Method 1 (σ_1^{prox}) is weaker than the rate we get for Method 1 (σ_1).

4.2. PCDM: Parallel Coordinate Descent Method

We will now compare Algorithm 1 with the Parallel Coordinate Descent Method (PCDM) of Richtárik and Takáč (Richtárik & Takáč, 2015), which can also be applied to problem (19).

Algorithm 2 PCDM (Richtárik & Takáč, 2015)

```
1: Parameters: proper sampling $\hat{S}$; $v \in \mathbb{R}^n_{++}$
2: Initialization: choose initial point $x^0 \in \mathbb{R}^n$
3: for $k = 0, 1, 2, \dots$ do
4:     Generate a random set of blocks $S_k \sim \hat{S}$
5:     Compute for each $i \in S_k$
       $h_i^k = \arg\min_{h_i \in \mathbb{R}} e_i^\top \nabla f(x^k) h_i + \frac{v_i}{2}|h_i|^2 + \psi_i(x_i^k + h_i)$
6:     Update: $x^{k+1} := x^k + \sum_{i \in S_k} h_i^k e_i$
7: end for
```

Proposition 12. *Let $v \in \mathbb{R}^n_{++}$ be a vector satisfying (8). Then the output sequence $\{x^k\}_{k \geq 0}$ of Algorithm 2 satisfies*

$$\mathbb{E}[F(x^{k+1}) - F(x^*)] \leq (1 - \sigma_3^{prox})\, \mathbb{E}[F(x^k) - F(x^*)],$$

where

$$\sigma_3^{prox} := \lambda_{\min}\left[\mathbf{D}(p)\left(\mathbf{D}(v + \gamma)\right)^{-1}(\mathbf{D}(\gamma) + \mathbf{G})\right].$$

Proof. Sketch: The proof is a minor modification of the arguments in (Richtárik & Takáč, 2015). □

Applying Theorem 6 to $\mathbf{M} + \mathbf{D}(\gamma)$ and $\mathbf{G} + \mathbf{D}(\gamma)$, we see that the rate of linear (geometric) convergence of our method is better than that of PCDM.

Proposition 13. $\sigma_1^{prox} \geq \sigma_3^{prox}$.

5. Empirical Risk Minimization

We now turn our attention to the *empirical risk minimization* problem:

$$\min_{w \in \mathbb{R}^d} P(w) := \frac{1}{n}\sum_{i=1}^n \phi_i(a_i^\top w) + \lambda g(w). \qquad (20)$$

We assume that $g : \mathbb{R}^d \to \mathbb{R}$ is a 1-strongly convex function with respect to the L2 norm. We also assume that each loss function $\phi_i : \mathbb{R} \to \mathbb{R}$ is convex and $1/\gamma$-smooth (i.e., $|\phi'(a) - \phi'(b)| \leq (1/\gamma)|a - b|$ for all $a, b \in \mathbb{R}$). Each a_i is a d-dimensional vector and for ease of presentation we write $\mathbf{A} = [a_1, \dots, a_n] = \sum_{i=1}^n a_i e_i^\top$. Let g^* and $\{\phi_i^*\}_i$ be the Fenchel conjugate functions of g and $\{\phi_i\}_i$, respectively. In the case of g, for instance, we have $g^*(s) = \sup_{w \in \mathbb{R}^d} \langle w, s \rangle - g(w)$. The (Fenchel) dual problem of (20) can be written as:

$$\max_{\alpha \in \mathbb{R}^n} D(\alpha) := \tfrac{1}{n}\textstyle\sum_{i=1}^n - \phi_i^*(-\alpha_i) - \lambda g^*\left(\frac{1}{\lambda n}\mathbf{A}\alpha\right). \quad (21)$$

It is well known (Shalev-Shwartz & Zhang, 2013b) that the solution of (20) w^* can be recovered from the solution of (21) α^* through the relation $w^* = \nabla g^*(1/(\lambda n)\mathbf{A}\alpha^*)$. Hence one can apply proximal variants of Methods 1, 2 and 3, to the dual problem (21) and then recover accordingly the solution of (20). In particular, we obtain Algorithm 3 (SDNA) by applying the proximal variant of Method 1 (Algorithm 1) to the dual problem (21).

Algorithm 3 Stochastic Dual Newton Ascent (SDNA)

```
1: Parameters: proper nonvacuous sampling $\hat{S}$
2: Initialization: $\alpha^0 \in \mathbb{R}^n$; $\bar{\alpha}^0 = \frac{1}{\lambda n}\mathbf{A}\alpha^0$
3: for $k = 0, 1, 2, \dots$ do
4:     Primal update: $w^k = \nabla g^*(\bar{\alpha}^k)$
5:     Generate a random set of blocks $S_k \sim \hat{S}$
6:     Compute: $\Delta\alpha^k = \arg\min_{h \in \mathbb{R}^n}\{\langle(\mathbf{A}^\top w^k)_{S_k}, h\rangle + \frac{1}{2\lambda n} h^\top(\mathbf{A}^\top\mathbf{A})_{S_k} h + \sum_{i \in S_k}\phi_i^*(-\alpha_i^k - h_i)\}$
7:     Dual update: $\alpha^{k+1} := \alpha^k + (\Delta\alpha^k)_{S_k}$
8:     Average update: $\bar{\alpha}^{k+1} = \bar{\alpha}^k + \frac{1}{\lambda n}\sum_{i \in S_k}\Delta\alpha_i^k a_i$
9: end for
```

With each proper sampling $\hat{S}$ we associate the number:

$$\theta(\hat{S}) := \min_i \tfrac{p_i \lambda \gamma n}{v_i + \lambda \gamma n}, \qquad (22)$$

where $v = (v_1, \dots, v_n) \in \mathbb{R}^n_{++}$ is a vector satisfying:

$$\mathbb{E}[(\mathbf{A}^\top\mathbf{A})_{\hat{S}}] \preceq \mathbf{D}(p)\mathbf{D}(v). \qquad (23)$$

We can now state the main result of this section:

Theorem 14 (Complexity of SDNA)**.** *The output sequence $\{w^k, \alpha^k\}_{k \geq 0}$ of Algorithm 3 satisfies:*

$$\mathbb{E}[P(w^k) - D(\alpha^k)] \leq \frac{(1 - \sigma_1^{prox})^k}{\theta(\hat{S})}(D(\alpha^*) - D(\alpha^0)), \qquad (24)$$

where $\hat{\sigma}_1^{prox}$ is given by

$$\lambda_{\min}\left[\mathbf{D}(p)\left(\tfrac{1}{\lambda\gamma n}\mathbb{E}[(\mathbf{A}^\top\mathbf{A})_{\hat{S}}] + \mathbf{D}(p)\right)^{-1}\mathbf{D}(p)\right]. \quad (25)$$

When $\{\phi_i\}_i$ and g are quadratic functions, then (24) holds with $\hat{\sigma}_1^{prox}$ replaced by the following better rate:

$$\lambda_{\min}\left[\mathbb{E}\left[\left(\left(\tfrac{1}{\lambda n}\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I}\right)_{\hat{S}}\right)^{-1}\left(\tfrac{1}{\lambda n}\mathbf{A}^\top\mathbf{A} + \gamma\mathbf{I}\right)\right]\right]. \qquad (26)$$

When applying the proximal variant of Method 3 to the dual problem, i.e., replacing Line 6 in Algorithm 3 by:

$$\Delta\alpha^k = \arg\min_{h \in \mathbb{R}^n}\left\{\langle(\mathbf{A}^\top w^k)_{S_k}, h\rangle + \tfrac{1}{2\lambda n}h^\top(\mathbf{D}(v))_{S_k}h + \textstyle\sum_{i \in S_k}\phi_i^*(-\alpha_i^k - h_i)\right\},$$

where $v \in \mathbb{R}^n_+$ is a parameter satisfying (23), we obtain a new method which will be called Minibatch SDCA.

When only one dual coordinate is updated at each iteration ($\mathbb{P}(|\hat{S}| = 1) = 1$), both SDNA and Minibatch SDCA reduce to the proximal variant of SDCA (Shalev-Shwartz & Zhang, 2013b). The complexity of Minibatch SDCA is given in Theorem 15.

Theorem 15 (Complexity of Minibatch SDCA)**.** *The output sequence $\{w^k, \alpha^k\}_{k\geq 0}$ of Minibatch SDCA satisfies:*

$$\mathbb{E}[P(w^k) - D(\alpha^k)] \leq \frac{(1-\theta(\hat{S}))^k}{\theta(\hat{S})} \left(D(\alpha^*) - D(\alpha^0)\right).$$

Note that a *minibatch version of standard SDCA in the ERM setup has not been previously studied in the literature.* (Takáč et al., 2013) developed such a method but in the special case of hinge-loss (which is not smooth and hence does not fit our setup). (Shalev-Shwartz & Zhang, 2013a) studied minibatching but in conjunction with acceleration and the QUARTZ method of (Qu et al., 2014), which has been analyzed for an arbitrary sampling $\hat{S}$, uses a different primal update than SDNA. Hence, in order to compare SDNA with an SDCA-like method which is as close a match to SDNA as possible, it was necessary to develop a new method. *Theorem 15 is an extension of SDCA to allow it handle an arbitrary uniform sampling $\hat{S}$.*

We now compare the rates of SDNA and SDCA. The next result says that *the rate of SDNA is always superior to that of SDCA.* It can be derived directly from Proposition 13. Indeed, it can be verified that $\theta(\hat{S})$ and $\hat{\sigma}_1^{prox}$ can be obtained respectively from σ_3^{prox} and σ_1^{prox} with $\mathbf{M} = \frac{1}{\lambda n^2}\mathbf{A}^\top\mathbf{A}$, $\mathbf{G} = 0$ and $\gamma_i = \frac{\gamma}{n}$ for all i.

Theorem 16. *If $\hat{S}$ is a proper nonvacuous sampling, then $\theta(\hat{S}) \leq \hat{\sigma}_1^{prox}$.*

6. SDNA as Iterative Hessian Sketch

We now apply SDNA to the ridge regression problem:

$$\min_{w\in\mathbb{R}^d} \frac{1}{2n}\|\mathbf{A}^\top w - b\|^2 + \frac{\lambda}{2}\|w\|^2, \tag{27}$$

and show that the resulting primal update can be interpreted as an iterative Hessian sketch, alternative to the one proposed by (Pilanci & Wainwright, 2014). We first need to state a simple (and well known) duality result.

Lemma 17. *Let α^* be the optimal solution of*

$$\min_{\alpha\in\mathbb{R}^n} \frac{1}{2n}\|\alpha\|^2 - \frac{1}{n}\langle b, \alpha\rangle + \frac{1}{2\lambda n^2}\|\mathbf{A}\alpha\|^2, \tag{28}$$

then the optimal solution w^* of (27) is $w^* = \frac{1}{\lambda n}\mathbf{A}\alpha^*$.

Indeed, problem (27) is a special case of (20) for $g(w) \equiv \frac{1}{2}\|w\|^2$ and $\phi_i(a) \equiv \frac{1}{2}(a - b_i)^2$ for all $i \in [n]$. Problem (28) is the dual of (27) and the result follows from (21).

The interpretation of SDNA as a variant of the Iterative Hessian sketch method of (Pilanci & Wainwright, 2014) follows immediately from the following theorem.

Theorem 18. *The output sequence $\{w^k, \alpha^k\}_{k\geq 0}$ of Algorithm 3 applied on problem (27) satisfies:*

$$w^{k+1} = \arg\min_{w\in\mathbb{R}^d} \left\{ \frac{1}{2n}\|\mathbf{S}_k^\top(\mathbf{A}^\top w - b)\|^2 + \frac{\lambda}{2}\|w\|^2 + \left\langle w, \frac{1}{n}\mathbf{A}\mathbf{I}_{S_k}\alpha^k - \lambda w^k \right\rangle \right\}, \tag{29}$$

where $\mathbf{S}_k$ denotes the n-by-$|S_k|$ submatrix of the identity matrix $\mathbf{I}_n$ with columns in the random subset S_k.

7. Numerical Experiments

Experiment 1. In our first experiment (Figure 1) we compare SDNA and our new minibatch version of SDCA on two real (`mushrooms`: $d = 112$, $n = 8,124$; `cov`: $d = 54$, $n = 522,911$) and one synthetic ($d = 1,024$, $n = 2,048$) dataset. In both cases, we used $\lambda = 1/n$ as the regularization parameter and $g(w) = \frac{1}{2}\|w\|^2$.

As τ increases, SDNA requires less passes over data (epochs), while SDCA requires more passes over data. It can be shown that this behavior can be predicted from the complexity results for these two methods. The difference in performance depends on the choice of the dataset and can be quite dramatic (see the two plots on the right).

Experiment 2. In the second experiment (Figure 2), we investigate how much time it takes for the methods to process a single epoch, using the same datasets as before. As τ increases, SDNA does more work as the subproblems it needs to solve in each iteration involve a $\tau \times \tau$ submatrix of the Hessian of the smooth part of the dual objective function. On the other hand, the work SDCA needs to do is much smaller, and does nearly not increase with the minibatch size τ. This is because the subproblems are separable. As before, all experiments are done using a single core (however, both methods would benefit from a parallel implementation).

Experiment 3. Finally, in Figure 3 we put the insights gained from the previous two experiments together: we look at the performance of SDNA for various choices of τ by comparing runtime and duality gap error. We should expect that increasing τ would lead to a faster method in terms of passes over data, but that this would also lead to slower iterations. The question is, does the gain outweight the loss? The answer is: yes, for small enough minibatch sizes. Indeed, looking at Figure 3, we see that the runtime of SDNA improved up to the point $\tau = 16$ for the first two datasets (and up to $\tau = 64$ for `cov`), and then starts to deteriorate. In situations where it is costly to fetch data from

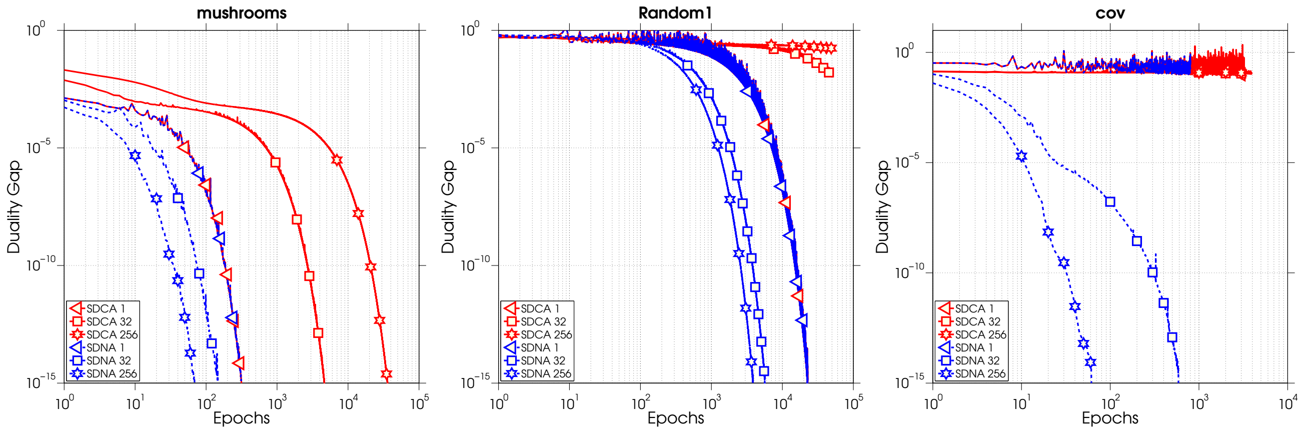

Figure 1. Comparison of SDNA and SDCA for minibatch sizes $\tau = 1, 32, 256$ on a real (left) and synthetic (right) dataset. The methods coincide for $\tau = 1$ (in theory and in practice).

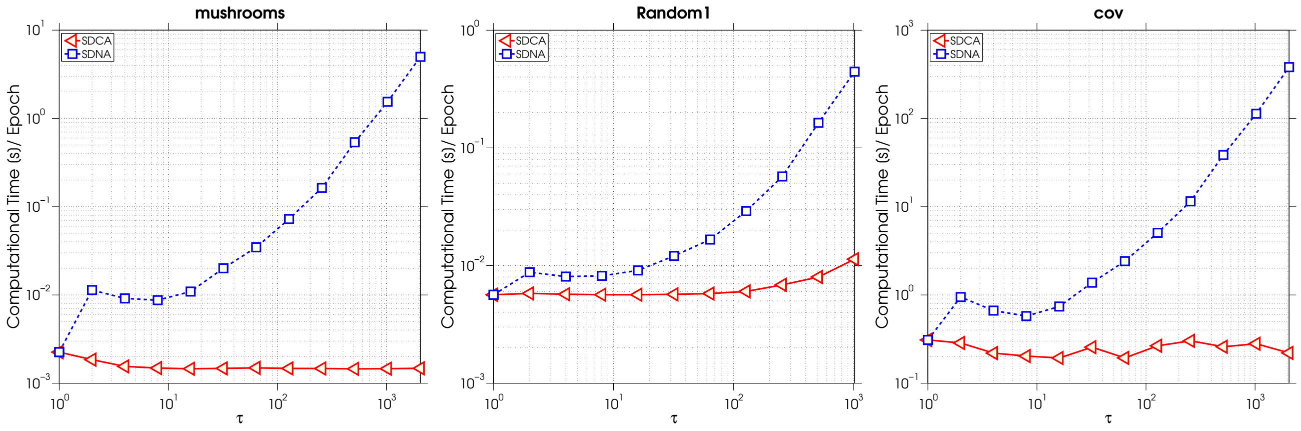

Figure 2. Processing time of one epoch for SDNA and SDCA as a function of the minibatch size τ.

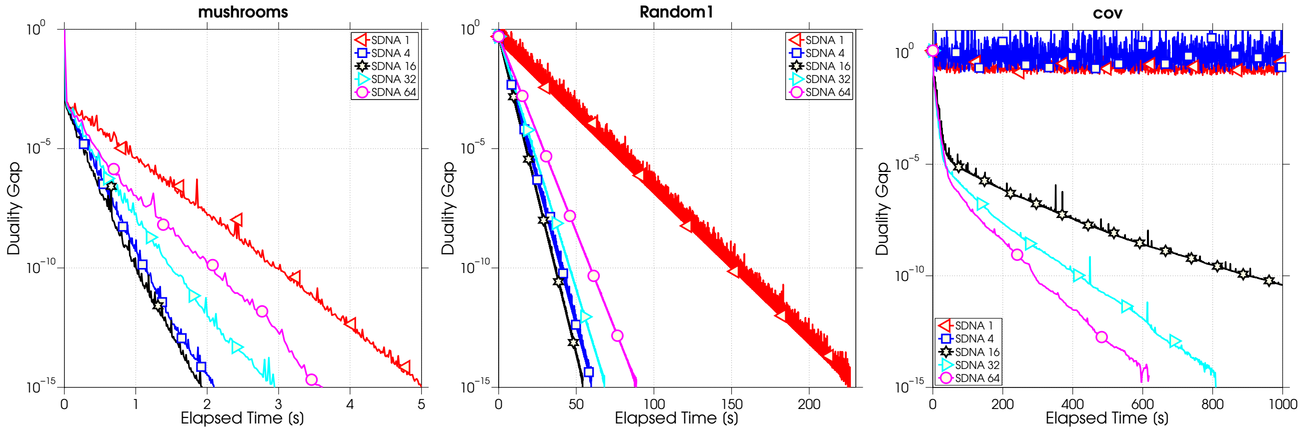

Figure 3. Runtime of SDNA for minibatch sizes $\tau = 1, 4, 16, 32, 64$.

memory to a (fast) processor, much larger minibatch sizes would be optimal.

Acknowledgements

Z. Qu and P. Richtárik would like to acknowledge support from the EPSRC Grant EP/K02325X/1, *Accelerated Coordinate Descent Methods for Big Data Optimization*, O. Fercoq would like to acknowledge support from the Orange/Telecom ParisTech think tank Phi-TAB.

References

Bordes, Antoine, Bottou, Léon, and Gallinari, Patrick. SGD-QN: Careful quasi-Newton stochastic gradient descent. *JMLR*, 10:1737–1754, 2009.

Byrd, R.H., Hansen, S.L., Nocedal, Jorge, and Singer, Yoram. A stochastic quasi-Newton method for large-scale optimization. *arXiv:1401.7020*, 2014.

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pp. 1646–1654, 2014.

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

Fercoq, Olivier and Richtárik, Peter. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

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. A flexible coordinate descent method for big data applications. *arXiv preprint arXiv:1507.03713*, 2015.

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, Microsoft Research, July 2014.

Mairal, Julien. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

Pilanci, Mert and Wainwright, Martin J. Iterative Hessian sketch: fast and accurate solution approximation for constrained least-squares. *arXiv:1411.0347*, 2014.

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

Qu, Zheng and Richtárik, Peter. Coordinate descent methods with arbitrary sampling II: Expected separable overapproximation. *arXiv:1412.8063*, 2014b.

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*, 2013.

Richtárik, Peter and Takáč, Martin. On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters*, pp. 1–11, 2015.

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.

Richtárik, Peter and Takáč, Martin. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, pp. 1–52, 2015. ISSN 0025-5610. doi: 10.1007/s10107-015-0901-6.

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

Schraudolph, Nicol N., Yu, Jin, and Günter, Simon. A stochastic quasi-Newton method for online convex optimization. In *AISTATS*, pp. 433–440, 2007.

Shalev-Shwartz, Shai and Ben-David, Shai. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Shalev-Shwartz, Shai and Zhang, Tong. Accelerated mini-batch stochastic dual coordinate ascent. In *NIPS*, pp. 378–385, 2013a.

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

Shalev-Shwartz, Shai, Singer, Yoram, Srebro, Nati, and Cotter, Andrew. Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical Programming*, pp. 3–30, 2011.

Sohl-Dickstein, Jascha, Poole, Ben, and Ganguli, Surya. Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods. In *ICML*, 2014.

Takáč, Martin, Bijral, Avleen, Richtárik, Peter, and Srebro, Nathan. Mini-batch primal and dual methods for SVMs. In *ICML*, 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.

Tseng, Paul and Yun, Sangwoon. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2009.

Xiao, Lin and Zhang, Tong. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

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