Randomized Block Cubic Newton Method

Nikita Doikov [1] **Peter Richtárik** [2 3 4]

Abstract

We study the problem of minimizing the sum of three convex functions—a differentiable, twice-differentiable and a non-smooth term—in a high dimensional setting. To this effect we propose and analyze a randomized block cubic Newton (RBCN) method, which in each iteration builds a model of the objective function formed as the sum of the *natural* models of its three components: a linear model with a quadratic regularizer for the differentiable term, a quadratic model with a cubic regularizer for the twice differentiable term, and perfect (proximal) model for the nonsmooth term. Our method in each iteration minimizes the model over a random subset of blocks of the search variable. RBCN is the first algorithm with these properties, generalizing several existing methods, matching the best known bounds in all special cases. We establish $\mathcal{O}(1/\epsilon)$, $\mathcal{O}(1/\sqrt{\epsilon})$ and $\mathcal{O}(\log(1/\epsilon))$ rates under different assumptions on the component functions. Lastly, we show numerically that our method outperforms the state of the art on a variety of machine learning problems, including cubically regularized least-squares, logistic regression with constraints, and Poisson regression.

1. Introduction

In this paper we develop an efficient randomized algorithm for solving an optimization problem of the form

$$\min_{x \in Q} F(x) \stackrel{\text{def}}{=} g(x) + \phi(x) + \psi(x), \qquad (1)$$

where $Q \subseteq \mathbb{R}^N$ is a closed convex set, and g, ϕ and ψ are convex functions with different smoothness and structural properties. Our aim is to capitalize on these different properties in the design of our algorithm. We assume that g has Lipschitz gradient[1], ϕ has Lipschitz Hessian, while ψ is allowed to be nonsmooth, albeit "simple".

1.1. Block Structure

Moreover, we assume that the N coordinates of x are partitioned into n blocks of sizes $N_1, \ldots, N_n$, with $\sum_i N_i = N$, and then write $x = (x_{(1)}, \ldots, x_{(n)})$, where $x_{(i)} \in \mathbb{R}^{N_i}$. This block structure is typically dictated by the particular application considered. Once the block structure is fixed, we further assume that ϕ and ψ are *block separable*. That is, $\phi(x) = \sum_{i=1}^n \phi_i(x_{(i)})$ and $\psi(x) = \sum_{i=1}^n \psi_i(x_{(i)})$, where ϕ_i are twice differentiable with Lipschitz Hessians, and ψ_i are closed convex (and possibly nonsmooth) functions.

Revealing this block structure, problem (1) takes the form

$$\min_{x \in Q} F(x) \stackrel{\text{def}}{=} g(x) + \sum_{i=1}^n \phi_i(x_{(i)}) + \sum_{i=1}^n \psi_i(x_{(i)}). \qquad (2)$$

We are specifically interested in the case when n is *big*, in which case it make sense to update a small number of the block in each iteration only.

1.2. Related Work

There has been a substantial and growing volume of research related to second-order and block-coordinate optimization. In this part we briefly mention some of the papers most relevant to the present work.

A major leap in second-order optimization theory was made since the cubic Newton method was proposed by Griewank (1981) and independently rediscovered by Nesterov & Polyak (2006), who also provided global complexity guarantees.

Cubic regularization was equipped with acceleration by Nesterov (2008), adaptive stepsizes by (Cartis et al., 2011a;b) and extended to a universal framework by Grapiglia & Nesterov (2017). The universal schemes can automatically

[1]National Research University Higher School of Economics, Samsung-HSE Laboratory, Moscow, Russia [2]King Abdullah University of Science and Technology, Thuwal, Saudi Arabia [3]University of Edinburgh, Edinburgh, United Kingdom [4]Moscow Institute of Physics and Technology, Dolgoprudny, Russia. Correspondence to: Nikita Doikov <nikita.doikov@gmail.com>, Peter Richtárik <peter.richtarik@kaust.edu.sa, peter.richtarik@ed.ac.uk>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

[1]Our assumption is bit more general than this; see Assumptions 1, 2 for details.

adjust to the implicit smoothness level of the objective. Cubically regularized second-order schemes for solving systems of nonlinear equations were developed by Nesterov (2007) and randomized variants for stochastic optimization were considered by Tripuraneni et al. (2017); Ghadimi et al. (2017); Kohler & Lucchi (2017); Cartis & Scheinberg (2018).

Despite their attractive global iteration complexity guarantees, the weakness of second-order methods in general, and cubic Newton in particular, is their high computational cost per iteration. This issue remains the subject of active research. For successful theoretical results related to the approximation of the cubic step we refer to (Agarwal et al., 2016) and (Carmon & Duchi, 2016).

At the same time, there are many successful attempts to use *block coordinate* randomization to accelerate first-order (Tseng & Yun, 2009; Richtárik & Takáč, 2014; 2016) and second-order (Qu et al., 2016; Mutnỳ & Richtárik, 2018) methods.

In this work we are addressing the issue of combining block-coordinate randomization with cubic regularization, to get *a second-order method with proven global complexity guarantees and with a low cost per iteration*.

A powerful advance in convex optimization theory was the advent of *composite* or *proximal* first-order methods (see (Nesterov, 2013) as a modern reference). This technique has become available as an algorithmic tool in block coordinate setting as well (Richtárik & Takáč, 2014; Qu et al., 2016). Our aim in this work is the development of a *composite cubically regularized second-order method*.

1.3. Contributions

We propose a new randomized second-order proximal algorithm for solving convex optimization problems of the form (2). Our method, *Randomized Block Cubic Newton (RBCN)* (see Algorithm 1) treats the three functions appearing in (1) differently, according to their nature.

Our method is a *randomized block method* because in each iteration we update a random subset of the n blocks only. This facilitates faster convergence, and is suited to problems where n is very large. Our method is *proximal* because we keep the functions ψ_i in our model, which is minimized in each iteration, without any approximation. Our method is a *cubic Newton* method because we approximate each ϕ_i using a cubically-regularized second order model.

We are not aware of *any method* that can solve (2) via using the most appropriate models of the three functions (quadratic with a constant Hessian for g, cubically regularized quadratic for ϕ and no model for ψ), not even in the case $n = 1$.

Our approach generalizes several existing results:

- In the case when $n = 1$, $g = 0$ and $\psi = 0$, RBCN reduces to the cubically-regularized Newton method of Nesterov & Polyak (2006). Even when $n = 1$, RBCN can be seen as an extension of this method to *composite* optimization. For $n > 1$, RBCN provides an extension of the algorithm in Nesterov & Polyak (2006) to the *randomized block coordinate* setting, popular for high-dimensional problems.
- In the special case when $\phi = 0$ and $N_i = 1$ for all i, RBCN specializes to the stochastic Newton (SN) method of Qu et al. (2016). Applied to the empirical risk minimization problem (see Section 7), our method has a dual interpretation (see Algorithm 2). In this case, our method reduces to the stochastic dual Newton ascent method (SDNA) also described in (Qu et al., 2016). Hence, RBCN can be seen as an extension of SN and SDNA to blocks of arbitrary sizes, and to the inclusion of the twice differentiable term ϕ.
- In the case when $\phi = 0$ and the simplest over approximation of g is assumed: $0 \preceq \nabla^2 g(x) \preceq LI$, the composite block coordinate gradient method Tseng & Yun (2009) can be applied to solve (1). Our method extends this in two directions: we add twice-differentiable terms ϕ, and use a tighter model for g, using all global curvature information (if available).

We prove high probability global convergence guarantees under several regimes, summarized next:

- Under no additional assumptions on g, ϕ and ψ beyond convexity (and either boundedness of Q, or boundedness of the level sets of F on Q), we prove the rate

$$\mathcal{O}\left(\frac{n}{\tau\epsilon}\right),$$

  where τ is the mini-batch size (see Theorem 1).
- Under certain conditions combining the properties of g with the way the random blocks are sampled, formalized by the assumption $\beta > 0$ (see (12) for the definition of β), we obtain the rate

$$\mathcal{O}\left(\frac{n}{\tau \max\{1,\beta\}\sqrt{\epsilon}}\right)$$

  (see Theorem 2). In the special case when $n = 1$, we necessarily have $\tau = 1$ and $\beta = \mu/L$ (reciprocal of the condition number of g) we get the rate $\mathcal{O}(\frac{L}{\mu\sqrt{\epsilon}})$. If g is quadratic and $\tau = n$, then $\beta = 1$ and the resulting complexity $\mathcal{O}(\frac{1}{\sqrt{\epsilon}})$ recovers the rate of cubic Newton established by Nesterov & Polyak (2006).

- Finally, if g is strongly convex, the above result can be improved (see Theorem 3) to

$$\mathcal{O}\left(\frac{n}{\tau \max\{1,\beta\}} \log \frac{1}{\epsilon}\right).$$

1.4. Contents

The rest of the paper is organized as follows. In Section 2 we introduce the notation and elementary identities needed to efficiently handle the block structure of our model. In Section 3 we make the various smoothness and convexity assumptions on g and ϕ_i formal. Section 4 is devoted to the description of the block sampling process used in our method, along with some useful identities. In Section 5 we describe formally our randomized block cubic Newton (RBCN) method. Section 6 is devoted to the statement and description of our main convergence results, summarized in the introduction. Missing proofs are provided in the supplementary material. In Section 7 we show how to apply our method to the empirical risk minimization problem. Applying RBCN to its dual leads to Algorithm 2. Finally, our numerical experiments on synthetic and real datasets are described in Section 8.

2. Block Structure

To model a block structure, we decompose the space $\mathbb{R}^N$ into n subspaces in the following standard way. Let $\mathbf{U} \in \mathbb{R}^{N\times N}$ be a column permutation of the $N \times N$ identity matrix $\mathbf{I}$ and let a decomposition $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_n]$ be given, where $\mathbf{U}_i \in \mathbb{R}^{N\times N_i}$ are n submatrices, $N = \sum_{i=1}^n N_i$. Subsequently, any vector $x \in \mathbb{R}^N$ can be uniquely represented as $x = \sum_{i=1}^n \mathbf{U}_i x_{(i)}$, where $x_{(i)} \stackrel{\text{def}}{=} \mathbf{U}_i^T x \in \mathbb{R}^{N_i}$.

In what follows we will use the standard Euclidean inner product: $\langle x, y\rangle \stackrel{\text{def}}{=} \sum_i x_i y_i$, Euclidean norm of a vector: $\|x\| \stackrel{\text{def}}{=} \sqrt{\langle x, x\rangle}$ and induced spectral norm of a matrix: $\|\mathbf{A}\| \stackrel{\text{def}}{=} \max_{\|x\|=1} \|\mathbf{A}x\|$. Using block decomposition, for two vectors $x, y \in \mathbb{R}^N$ we have:

$$\langle x, y\rangle = \left\langle \sum_{i=1}^n \mathbf{U}_i x_{(i)}, \sum_{j=1}^n \mathbf{U}_j y_{(j)} \right\rangle = \sum_{i=1}^n \langle x_{(i)}, y_{(i)}\rangle.$$

For a given nonempty subset S of $[n] \stackrel{\text{def}}{=} \{1, \dots, n\}$ and for any vector $x \in \mathbb{R}^N$ we denote by $x_{[S]} \in \mathbb{R}^N$ the vector obtained from x by retaining only blocks $x_{(i)}$ for which $i \in S$ and zeroing all other:

$$x_{[S]} \stackrel{\text{def}}{=} \sum_{i\in S} \mathbf{U}_i x_{(i)} = \sum_{i\in S} \mathbf{U}_i \mathbf{U}_i^T x.$$

Furthermore, for any matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ we write $\mathbf{A}_{[S]} \in \mathbb{R}^{N\times N}$ for the matrix obtained from $\mathbf{A}$ by retaining only elements whose indices are both in some coordinate blocks from S, formally:

$$\mathbf{A}_{[S]} \stackrel{\text{def}}{=} \left(\sum_{i\in S} \mathbf{U}_i \mathbf{U}_i^T\right) \mathbf{A} \left(\sum_{i\in S} \mathbf{U}_i \mathbf{U}_i^T\right).$$

Note that these definitions imply that

$$\langle \mathbf{A}_{[S]} x, y\rangle = \langle \mathbf{A} x_{[S]}, y_{[S]}\rangle, \quad x, y \in \mathbb{R}^N.$$

Next, we define the *block-diagonal* operator, which, up to permutation of coordinates, retains diagonal blocks and nullifies the off-diagonal blocks:

$$\text{blockdiag}(\mathbf{A}) \stackrel{\text{def}}{=} \sum_{i=1}^n \mathbf{U}_i \mathbf{U}_i^T \mathbf{A} \mathbf{U}_i \mathbf{U}_i^T = \sum_{i=1}^n \mathbf{A}_{[\{i\}]}.$$

Finally, denote $\mathbb{R}^N_{[S]} \stackrel{\text{def}}{=} \left\{x_{[S]} \,|\, x \in \mathbb{R}^N\right\}$. This is a linear subspace of $\mathbb{R}^N$ composed of vectors which are zero in blocks $i \notin S$.

3. Assumptions

In this section we formulate our main assumptions about differentiable components of (2) and provide some examples to illustrate the concepts.

We assume that $g : \mathbb{R}^N \to \mathbb{R}$ is a differentiable function and all $\phi_i : \mathbb{R}^{N_i} \to \mathbb{R}$, $i \in [n]$ are twice differentiable. Thus, at any point $x \in \mathbb{R}^N$ we should be able to compute all the gradients $\{\nabla g(x), \nabla \phi_1(x_{(1)}), \dots, \nabla \phi_n(x_{(n)})\}$ and the Hessians $\{\nabla^2 \phi_1(x_{(1)}), \dots, \nabla^2 \phi_n(x_{(n)})\}$, or at least their actions on arbitrary vector h of appropriate dimension.

Next, we formalize our assumptions about convexity and level of smoothness. Speaking informally, g is similar to a quadratic, and functions ϕ_i are arbitrary twice-differentiable and smooth.

Assumption 1 (Convexity) *There is a positive semidefinite matrix $\mathbf{G} \succeq 0$ such that for all $x, h \in \mathbb{R}^N$:*

$$g(x+h) \geq g(x) + \langle \nabla g(x), h\rangle + \frac{1}{2}\langle \mathbf{G}h, h\rangle, \qquad (3)$$

$$\phi_i(x_{(i)} + h_{(i)}) \geq \phi_i(x_{(i)}) + \langle \nabla \phi_i(x_{(i)}), h_{(i)}\rangle, \quad i \in [n].$$

In the special case when $\mathbf{G} = 0$, (3) postulates *convexity*. For positive definite $\mathbf{G}$, the objective will be *strongly convex* with the strong convexity parameter $\mu \stackrel{\text{def}}{=} \lambda_{\min}(\mathbf{G}) > 0$.

Note that for all ϕ_i we only require convexity. However, if we happen to know that any ϕ_i is strongly convex ($\lambda_{\min}(\nabla^2 \phi_i(y)) \geq \mu_i > 0$ for all $y \in \mathbb{R}^{N_i}$), we can *move* this strong convexity to g by subtracting $\frac{\mu_i}{2}\|x_{(i)}\|^2$ from ϕ_i and adding it to g. This extra knowledge may in some particular cases improve convergence guarantees for our algorithm, but does not change the actual computations.

Assumption 2 (Smoothness of g) *There is a positive semidefinite matrix $\mathbf{A} \succeq 0$ such that for all $x, h \in \mathbb{R}^N$:*

$$g(x+h) \leq g(x) + \langle \nabla g(x), h \rangle + \frac{1}{2}\langle \mathbf{A}h, h \rangle. \quad (4)$$

The main example of g is a quadratic function $g(x) = \frac{1}{2}\langle \mathbf{M}x, x \rangle$ with a symmetric positive semidefinite $\mathbf{M} \in \mathbb{R}^{N \times N}$ for which both (3) and (4) hold with $\mathbf{G} = \mathbf{A} = \mathbf{M}$.

Of course, any convex g with Lipschitz-continuous gradient with a constant $L \geq 0$ satisfies (3) and (4) with $\mathbf{G} = 0$ and $\mathbf{A} = L\mathbf{I}$ (Nesterov, 2004).

Assumption 3 (Smoothness of ϕ_i) *For every $i \in [n]$ there is a nonnegative constant $\mathcal{H}_i \geq 0$ such that the Hessian of ϕ_i is Lipschitz-continuous:*

$$\|\nabla^2 \phi_i(x+h) - \nabla^2 \phi_i(x)\| \leq \mathcal{H}_i \|h\|, \quad (5)$$

for all $x, h \in \mathbb{R}^{N_i}$.

Examples of functions which satisfy (5) with a known Lipschitz constant of Hessian $\mathcal{H}$ are *quadratic*: $\phi(t) = \|Ct - t_0\|^2$ ($\mathcal{H} = 0$ for all the parameters), *cubed norm*: $\phi(t) = (1/3)\|t - t_0\|^3$ ($\mathcal{H} = 2$, see Lemma 5 in (Nesterov, 2008)), *logistic regression loss*: $\phi(t) = \log(1 + \exp(t))$ ($\mathcal{H} = 1/(6\sqrt{3})$, see Proposition 1 in the supplementary material).

For a fixed set of indices $S \subset [n]$ denote

$$\phi_S(x) \stackrel{\text{def}}{=} \sum_{i \in S} \phi_i(x_{(i)}), \quad x \in \mathbb{R}^N.$$

Then we have:

$$\langle \nabla \phi_S(x), h \rangle = \sum_{i \in S} \langle \nabla \phi_i(x_{(i)}), h_{(i)} \rangle, \quad x, h \in \mathbb{R}^N,$$
$$\langle \nabla^2 \phi_S(x) h, h \rangle = \sum_{i \in S} \langle \nabla^2 \phi_i(x_{(i)}) h_{(i)}, h_{(i)} \rangle, \quad x, h \in \mathbb{R}^N.$$

Lemma 1 *If Assumption 3 holds, then for all $x, h \in \mathbb{R}^N$ we have the following second-order approximation bound:*

$$\left| \phi_S(x+h) - \phi_S(x) - \langle \nabla \phi_S(x), h \rangle - \frac{1}{2}\langle \nabla^2 \phi_S(x) h, h \rangle \right| \leq \max_{i \in S} \{\mathcal{H}_i\} \cdot \|h_{[S]}\|^3. \quad (6)$$

From now on we denote $\mathcal{H}_F \stackrel{\text{def}}{=} \max\{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_n\}$.

4. Sampling of Blocks

In this section we summarize some basic properties of sampling $\hat{S}$, which is a random set-valued mapping with values being subsets of $[n]$. For a fixed block-decomposition, with each sampling $\hat{S}$ we associate the *probability matrix* $\mathbf{P} \in \mathbb{R}^{N \times N}$ as follows: an element of $\mathbf{P}$ is the probability of choosing a pair of blocks which contains indices of this element. Denoting by $\mathbf{E} \in \mathbb{R}^{N \times N}$ the matrix of all ones, we have $\mathbf{P} = \mathbb{E}\left[\mathbf{E}_{[\hat{S}]}\right]$. Wel restrict our analysis to *uniform* samplings, defined next.

Assumption 4 (Uniform sampling) *Sampling $\hat{S}$ is uniform, i.e., $\mathbb{P}(i \in \hat{S}) = \mathbb{P}(j \in \hat{S}) \stackrel{\text{def}}{=} p$, for all $i, j \in [n]$.*

The above assumpotion means that the diagonal of $\mathbf{P}$ is constant: $\mathbf{P}_{ii} = p$ for all $i \in [N]$. It is easy to see that (Corollary 3.1 in (Qu & Richtárik, 2016)):

$$\mathbb{E}\left[\mathbf{A}_{[\hat{S}]}\right] = \mathbf{A} \circ \mathbf{P}, \quad (7)$$

where $\circ$ denotes the Hadamard product.

Denote $\tau \stackrel{\text{def}}{=} \mathbb{E}[|\hat{S}|] = np$ (expected minibatch size). The special uniform sampling defined by picking from all subsets of size τ uniformly at random is called *τ-nice sampling*. If $\hat{S}$ is τ-nice, then (see Lemma 4.3 in (Qu & Richtárik, 2016))

$$\mathbf{P} = \frac{\tau}{n}\left((1-\gamma)\,\mathrm{blockdiag}(\mathbf{E}) + \gamma \mathbf{E}\right), \quad (8)$$

where $\gamma = (\tau - 1)/(n-1)$.

In particular, the above results in the following:

Lemma 2 *For the τ-nice sampling $\hat{S}$, we have*

$$\mathbb{E}[\mathbf{A}_{[\hat{S}]}] = \frac{\tau}{n}\left(1 - \frac{\tau - 1}{n-1}\right)\mathrm{blockdiag}(\mathbf{A}) + \frac{\tau(\tau-1)}{n(n-1)}\mathbf{A}.$$

Proof: Combine (7) and (8). ■

5. Algorithm

Due to the problem structure (2) and utilizing the smoothness of the components (see (4) and (5)), for a fixed subset of indices $S \subset [n]$ it is natural to consider the following *model* of our objective F around a point $x \in \mathbb{R}^N$:

$$M_{H,S}(x; y) \stackrel{\text{def}}{=} F(x) + \langle (\nabla g(x))_{[S]}, y \rangle + \frac{1}{2}\langle \mathbf{A}_{[S]} y, y \rangle + \\ + \langle (\nabla \phi(x))_{[S]}, y \rangle + \frac{1}{2}\langle (\nabla^2 \phi(x))_{[S]} y, y \rangle + \frac{H}{6}\|y_{[S]}\|^3 + \\ + \sum_{i \in S}\Big(\psi_i(x_{(i)} + y_{(i)}) - \psi_i(x_{(i)})\Big). \quad (9)$$

The above model arises as a combination of a first-order model for g with global curvature information provided by matrix $\mathbf{A}$, second-order model with cubic regularization (following (Nesterov & Polyak, 2006)) for ϕ, and perfect model for the non-differentiable terms ψ_i (i.e., we keep these terms as they are).

Combining (4) and (6), and for large enough H ($H \geq \max_{i\in S}\mathcal{H}_i$ is sufficient), we get the global upper bound

$$F(x+y) \leq M_{H,S}(x;y), \quad x\in\mathbb{R}^N,\ y\in\mathbb{R}^N_{[S]}.$$

Moreover, the value of all summands in $M_{H,S}(x;y)$ depends on the subset of blocks $\{y_{(i)}|i\in S\}$ only, and therefore

$$T_{H,S}(x) \stackrel{\text{def}}{=} \operatorname*{argmin}_{\substack{y\in\mathbb{R}^N_{[S]}\\ \text{subject to } x+y\in Q}} M_{H,S}(x;y) \tag{10}$$

can be computed efficiently for small $|S|$ and as long as Q is *simple* (for example, affine). Denote the minimum of the cubic model by $M^*_{H,S}(x) \stackrel{\text{def}}{=} M_{H,S}(x;T_{H,S}(x))$. The RBCN method performs the update $x \leftarrow x + T_{H,S}(x)$, and is formalized as Algorithm 1.

Algorithm 1 RBCN: Randomized Block Cubic Newton

1: **Parameters:** sampling distribution $\hat{S}$
2: **Initialization:** choose initial point $x^0 \in Q$
3: **for** $k = 0, 1, 2, \ldots$ **do**
4: Sample $S_k \sim \hat{S}$
5: Find $H_k \in (0,\ 2\mathcal{H}_F]$ such that
$F(x^k + T_{H_k,S_k}(x^k)) \leq M^*_{H_k,S_k}(x^k)$
6: Make the step $x^{k+1} \stackrel{\text{def}}{=} x^k + T_{H_k,S_k}(x^k)$
7: **end for**

6. Convergence Results

In this section we establish several convergence rates for Algorithm 1 under various structural assumptions: for the general class of convex problems, and for the more specific strongly convex case. We will focus on the family of *uniform samplings* only, but generalizations to other sampling distributions are also possible.

6.1. Convex Loss

We start from the general situation where the term $g(x)$ and all the $\phi_i(x_{(i)})$ and $\psi_i(x_{(i)}), i\in[n]$ are convex, but not necessary strongly convex.

Denote by D the maximum distance from an optimum point x^* to the initial level set:

$$D \stackrel{\text{def}}{=} \sup\Big\{\|x-x^*\| \;\Big|\; x\in Q,\ F(x)\leq F(x^0)\Big\}.$$

Theorem 1 *Let Assumptions 1, 2, 3, 4 hold. Let solution $x^*\in Q$ of problem (1) exist, and assume the level sets are bounded: $D<+\infty$. Choose required accuracy $\varepsilon>0$ and confidence level $\rho\in(0,1)$. Then after*

$$K \geq \frac{2}{\varepsilon}\frac{n}{\tau}\left(1+\log\frac{1}{\rho}\right)\max\Big\{LD^2+\mathcal{H}_F D^3,\ F(x^0)-F^*\Big\} \tag{11}$$

iterations of Algorithm 1, where $L \stackrel{\text{def}}{=} \lambda_{\max}(\mathbf{A})$, we have

$$\mathbb{P}\Big(F(x^K)-F^*\leq\varepsilon\Big) \geq 1-\rho.$$

Given theoretical result provides global sublinear rate of convergence, with iteration complexity of the order $O\big(1/\varepsilon\big)$.

Note that for a case $\phi(x)\equiv 0$ we can put $\mathcal{H}_F=0$, and Theorem 1 in this situation restates well-known result about convergence of composite gradient-type block-coordinate methods (see, for example, (Richtárik & Takáč, 2014)).

6.2. Strongly Convex Loss

Here we study the case when the matrix $\mathbf{G}$ from the convexity assumption (3) is strictly positive definite: $\mathbf{G}\succ 0$, which means that the objective F is *strongly convex* with a constant $\mu \stackrel{\text{def}}{=} \lambda_{\min}(\mathbf{G}) > 0$.

Denote by β a *condition number* for the function g and sampling distribution $\hat{S}$: the maximum nonnegative real number such that

$$\beta\cdot\mathbb{E}_{S\sim\hat{S}}\big[\mathbf{A}_{[S]}\big] \preceq \frac{\tau}{n}\mathbf{G}. \tag{12}$$

If (12) holds for *all* nonnegative β we put by definition $\beta\equiv+\infty$.

A simple lower bound exists: $\beta\geq\frac{\mu}{L}>0$, where $L=\lambda_{\max}(\mathbf{A})$, as in Theorem 1. However, because (12) depends not only on g, but also on sampling distribution $\hat{S}$, it is possible that $\beta>0$ even if $\mu=0$ (for example, $\beta=1$ if $\mathbb{P}(S=[n])=1$ and $\mathbf{A}=\mathbf{G}\neq 0$).

The following theorems describe global iteration complexity guarantees of the order $O\big(1/\sqrt{\varepsilon}\big)$ and $O\big(\log(1/\varepsilon)\big)$ for Algorithm 1 in the cases $\beta>0$ and $\mu>0$ correspondingly, which is an improvement of general $O\big(1/\varepsilon\big)$.

Theorem 2 *Let Assumptions 1, 2, 3, 4 hold. Let solution $x^*\in Q$ of problem (1) exist, let level sets be bounded: $D<+\infty$, and assume that β, which is defined by (12), is greater than zero. Choose required accuracy $\varepsilon>0$ and confidence level $\rho\in(0,1)$. Then after*

$$K \geq \frac{2}{\sqrt{\varepsilon}}\frac{n}{\tau}\frac{1}{\sigma}\left(2+\log\frac{1}{\rho}\right)\sqrt{\max\Big\{\mathcal{H}_F D^3,\ F(x^0)-F^*\Big\}} \tag{13}$$

iterations of Algorithm 1, where $\sigma \stackrel{\text{def}}{=} \min\{\beta,1\}>0$, we have

$$\mathbb{P}\Big(F(x^K)-F^*\leq\varepsilon\Big) \geq 1-\rho.$$

Theorem 3 *Let Assumptions 1, 2, 3, 4 hold. Let solution $x^*\in Q$ of problem (1) exist, and assume that $\mu \stackrel{\text{def}}{=} \lambda_{\min}(\mathbf{G})$*

is strictly positive. Then after

$$K \geq \frac{3}{2} \log\left(\frac{F(x^0) - F^*}{\varepsilon\rho}\right) \frac{n}{\tau} \frac{1}{\sigma} \sqrt{\max\left\{\frac{\mathcal{H}_F D}{\mu}, 1\right\}} \quad (14)$$

iterations of Algorithm 1, we have

$$\mathbb{P}\Big(F(x^K) - F^* \leq \varepsilon\Big) \;\geq\; 1 - \rho.$$

Given complexity estimates show which parameters of the problem directly affect on the convergence of the algorithm.

Bound (13) improves initial estimate (11) by the factor $\sqrt{D_0/\varepsilon}$. The cost is additional term $\sigma^{-1} = (\min\{\beta, 1\})^{-1}$, which grows up while the condition number β becomes smaller.

Opposite and limit case is when the *quadratic* part of the objective is vanished ($g(x) \equiv 0 \Rightarrow \sigma = 1$). Algorithm 1 is turned to be a parallelized block-independent minimization of the objective components via cubically regularized Newton steps. Then, the complexity estimate coincides with a known result (Nesterov & Polyak, 2006) in a nonrandomized ($\tau = n,\ \rho \to 1$) setting.

Bound (14) guarantees a linear rate of convergence, which means logarithmic dependence on required accuracy ε for the number of iterations. The main complexity factor becomes a product of two terms: $\sigma^{-1} \cdot \max\{\mathcal{H}_F D/\mu, 1\}^{1/2}$. For the case $\phi(x) \equiv 0$ we can put $\mathcal{H}_F = 0$ and get the stochastic Newton method from (Qu et al., 2016) with its global linear convergence guarantee.

Despite the fact that linear rate is asymptotically better than sublinear, and $O\big(1/\sqrt{\varepsilon}\big)$ is asymptotically better than $O\big(1/\varepsilon\big)$, we need to take into account all the factors, which may slow down the algorithm. Thus, while $\mu = \lambda_{\min}(\mathbf{G}) \to 0$, estimate (13) is becoming better than (14), as well as (11) is becoming better than (13) while $\beta \to 0$.

6.3. Implementation Issues

Let us explain how one step of the method can be performed, which requires the minimization of the cubic model (10), possibly with some simple convex constraints.

The first and the classical approach was proposed in (Nesterov & Polyak, 2006) and, before for trust-region methods, in (Conn et al., 2000). It works with unconstrained ($Q \equiv \mathbb{R}^N$) and differentiable case ($\psi(x) \equiv 0$). Firstly we need to find a root of a special one-dimensional nonlinear equation (this can be done, for example, by simple Newton iterations). After that, we just solve one linear system to produce a step of the method. Then, total complexity of solving the subproblem can be estimated as $O(d^3)$ arithmetical operations, where d is the dimension of subproblem, in our case: $d = |S|$. Since some matrix factorization is used, the cost of the cubically regularized Newton step is actually similar by efficiency to the classical Newton one. See also (Gould et al., 2010) for detailed analysis. For the case of affine constraints, the same procedure can be applied. Example of using this technique is given by Lemma 3 from the next section.

Another approach is based on finding an inexact solution of the subproblem by the fast approximate eigenvalue computation (Agarwal et al., 2016) or by applying gradient descent (Carmon & Duchi, 2016). Both of these schemes provide global convergence guarantees. Additionally, they are Hessian-free. Thus we need only a procedure of multiplying quadratic part of (9) to arbitrary vector, without storing the full matrix. The latter approach is the most universal one and can be spread to the composite case, by using proximal gradient method or its accelerated variant (Nesterov, 2013).

There are basically two strategies to find parameter H_k on every iteration: a *constant choice* $H_k := \max_{i \in S_k}\{\mathcal{H}_i\}$ or $H_k := \mathcal{H}_F$, if Lipschitz constants of the Hessians are known, or simple *adaptive procedure*, which performs a truncated binary search and has a logarithmic cost per one step of the method. Example of such procedure can be found in primal-dual Algorithm 2 from the next section.

6.4. Extension of the Problem Class

The randomized cubic model (9), which has been considered and analyzed before, arises naturally from the separable structure (2) and by our smoothness assumptions (4), (5). Let us discuss an interpretation of Algorithm 1 in terms of general problem $\min\limits_{x \in \mathbb{R}^N} F(x)$ with twice-differentiable F (omitting non-differentiable component for simplicity). One can state and minimize the model $m_{H,S}(x; y) \equiv F(x) + \langle (\nabla F(x))_{[S]}, y\rangle + \frac{1}{2}\langle (\nabla^2 F(x))_{[S]} y, y\rangle + \frac{H}{6}\|y_{[S]}\|^3$ which is a *sketched* version of the originally proposed Cubic Newton method (Nesterov & Polyak, 2006). For alternative sketched variants of Newton-type methods but without cubic regularization see (Pilanci & Wainwright, 2015).

The latter model $m_{H,S}(x; y)$ is the same as $M_{H,S}(x; y)$ when inequality (4) from the smoothness assumption for g is exact equality, i.e. when the function g is a quadratic with the Hessian matrix $\nabla^2 g(x) \equiv \mathbf{A}$. Thus, we may use $m_{H,S}(x; y)$ instead of $M_{H,S}(x; y)$, which is still computationally cheap for small $|S|$. However, this model does not give any convergence guarantees for the general F, to the best of our knowledge, unless $S = [n]$. But it can be a workable approach, when the separable structure (2) is not provided.

Note also, that Assumption 3 about Lipschitz-continuity of the Hessian is not too restrictive. Recent result (Grapiglia

& Nesterov, 2017) shows that Newton-type methods with cubic regularization and with a standard procedure of *adaptive* estimation of $\mathcal{H}_F$ automatically fit the actual level of smoothness of the Hessian without any additional changes in the algorithm.

Moreover, step (10) of the method as the global minimum of $M_{H,S}(x;y)$ is well-defined and computationally tractable even in nonconvex cases (Nesterov & Polyak, 2006). Thus we can try to apply the method to nonconvex objective as well, but without known theoretical guarantees for $S \neq [n]$.

7. Empirical Risk Minimization

One of the most popular examples of optimization problems in machine learning is *empirical risk minimization* problem, which in many cases can be formulated as follows:

$$\min_{w\in\mathbb{R}^d}\left[P(w) \equiv \frac{1}{m}\sum_{i=1}^m \phi_i(b_i^T w) + \lambda g(w)\right], \qquad (15)$$

where ϕ_i are convex *loss functions*, g is a *regularizer*, variables w are *weights* of a model and m is a size of a dataset.

7.1. Constrained Problem Reformulation

Let us consider the case, when the dimension d of problem (15) is very *huge* and $d \gg m$. This asks us to use some coordinate-randomization technique. Note that formulation (15) does not directly fit our problem setup (2), but we can easily transform it to the following constrained optimization problem, by introducing new variables $\alpha_i \equiv b_i^T w$:

$$\min_{\substack{w\in\mathbb{R}^d\\ \alpha\in\mathbb{R}^m}}\left[\frac{1}{m}\sum_{i=1}^m \phi_i(\alpha_i)+\lambda g(w)+\sum_{i=1}^m \mathbb{I}\{\alpha_i = b_i^T w\}\right]. \quad (16)$$

Following our framework, on every step we will sample a random subset of coordinates $S \subset [d]$ of weights w, build the cubic model of the objective (assuming that ϕ_i and g satisfy (4), (5)):

$$M_{H,S}(w,\alpha;y,h) \equiv \lambda\left(\langle(\nabla g(w))_{[S]}, y\rangle + \frac{1}{2}\langle \mathbf{A}_{[S]}y, y\rangle\right)+$$
$$+\frac{1}{m}\left(\sum_{i=1}^m\left(\phi_i'(\alpha_i)h_i + \frac{1}{2}\phi_i''(\alpha_i)h_i^2\right) + \frac{H}{6}\|h\|^3\right) + P(w)$$

and minimize it by y and h on the affine set:

$$(y^*,h^*) := \operatorname*{argmin}_{\substack{y\in\mathbb{R}^d_{[S]},h\in\mathbb{R}^m\\ \text{subject to } h=\mathbf{B}y}} M_{H,S}(w,\alpha;y,h), \qquad (17)$$

where rows of matrix $\mathbf{B} \in \mathbb{R}^{m\times d}$ are b_i^T. Then, updates of the variables are: $w^+ := w + y^*$ and $\alpha^+ := \alpha + h^*$.

The following lemma is addressing the issue of how to solve (17), which is required on every step of the method. Its proof can be found in the supplementary material.

Lemma 3 *Denote by $\hat{\mathbf{B}} \in \mathbb{R}^{m\times|S|}$ the submatrix of $\mathbf{B}$ with row indices from S, by $\hat{\mathbf{A}} \in \mathbb{R}^{|S|\times|S|}$ the submatrix of $\mathbf{A}$ with elements whose both indices are from S, by $b_1 \in \mathbb{R}^{|S|}$ the subvector of $\nabla g(w)$ with element indices from S. Denote vector $b_2 \equiv \left(\phi_i'(\alpha_i)\right)_{i=1}^m$ and $b \equiv m\lambda b_1 + \hat{\mathbf{B}}^T b_2$. Define the family of matrices $\mathbf{Z}(\tau) : \mathbb{R}_+ \to \mathbb{R}^{|S|\times|S|}$:*

$$\mathbf{Z}(\tau) \stackrel{\text{def}}{=} m\lambda\hat{\mathbf{A}} + \hat{\mathbf{B}}^T\left(\operatorname{diag}(\phi_i''(\alpha_i)) + \frac{H\tau}{2}\mathbf{I}\right)\hat{\mathbf{B}}.$$

Then the solution (y^*,h^*) of (17) can be found from the equations: $\mathbf{Z}(\tau^*)y_S^* = -b$, $h^* = \hat{\mathbf{B}}y_S^*$, where $\tau^* \geq 0$ satisfies one-dimensional nonlinear equation: $\tau^* = \|\hat{\mathbf{B}}(\mathbf{Z}(\tau^*))^\dagger b\|$ and $y_S^* \in \mathbb{R}^{|S|}$ is the subvector of the solution y^* with element indices from S.

Thus, after we find the root of nonlinear *one-dimensional* equation, we need to solve $|S|\times|S|$ linear system to compute y^*. Then, to find h^* we do one matrix-vector multiplication with the matrix of size $m \times |S|$. Matrix $\mathbf{B}$ usually has a sparse structure when m is big, which also should be used in effective implementation.

7.2. Maximization of the Dual Problem

Another approach to solving optimization problem (15) is to maximize its Fenchel dual (Rockafellar, 1997):

$$\max_{\alpha\in\mathbb{R}^m}\left[D(\alpha) \equiv \frac{1}{m}\sum_{i=1}^m -\phi_i^*(-\alpha_i) - \lambda g^*\left(\frac{1}{\lambda m}\mathbf{B}^T\alpha\right)\right], \qquad (18)$$

where g^* and $\{\phi_i^*\}$ are *the Fenchel conjugate* functions of g and $\{\phi_i\}$ respectively, $f^*(s) \stackrel{\text{def}}{=} \sup_x[\,\langle s,x\rangle - f(x)\,]$ for arbitrary f. It is know (Bertsekas, 1978), that if ϕ_i is twice-differentiable in a neighborhood of y and $\nabla^2\phi_i(y) \succ 0$ in this area, then its Fenchel conjugate ϕ_i^* is also twice-differentiable in some neighborhood of $s = \nabla\phi_i(y)$ and it holds: $\nabla^2\phi_i^*(s) = (\nabla^2\phi_i(y))^{-1}$.

Then, in a case of smooth differentiable g^* and twice-differentiable $\phi_i^*, i \in [m]$ we can apply our framework to (18), by doing cubic steps in random subsets of the dual variables $\alpha \in \mathbb{R}^m$. The primal $w \in \mathbb{R}^d$ corresponded to particular α can be computed from the stationary equation

$$w = \nabla g^*\left(\frac{1}{\lambda m}\mathbf{B}^T\alpha\right),$$

which holds for solutions of primal (15) and dual (18) problems in a case of strong duality.

Let us assume that the function g is 1-strongly convex (which is of course true for ℓ_2-regularizer $1/2\|w\|_2^2$). Then for $G(\alpha) \equiv \lambda g^*\left(\frac{1}{\lambda m}\mathbf{B}^T\alpha\right)$ the uniform bound for the Hessian exists: $\nabla^2 G(\alpha) \preceq \frac{1}{\lambda m^2}\mathbf{B}^T\mathbf{B}$. As before we build the

randomized cubic model and compute its minimizer (setting $Q \equiv \bigcap_{i=1}^m \operatorname{dom} \phi_i^*$):

$$M_{H,S}(\alpha, h) \equiv -D(\alpha) + \lambda \left\langle \nabla g^* \left(\frac{1}{\lambda m} \mathbf{B}^T \alpha \right), h_{[S]} \right\rangle + \\ + \frac{1}{2\lambda m^2} \|\mathbf{B} h_{[S]}\|^2 + \frac{1}{m} \sum_{i \in S} \left[-\langle \nabla \phi_i^*(-\alpha_i), h_i \rangle + \right. \\ \left. + \frac{1}{2} \langle \nabla^2 \phi_i^*(-\alpha_i) h_i, h_i \rangle \right] + \frac{H}{6} \|h_{[S]}\|^3; \; S \subset [m],$$

$$T_{H,S}(\alpha) \equiv \operatorname*{argmin}_{\substack{h \in \mathbb{R}^m_{[S]} \\ \text{subject to } \alpha + h \in Q}} M_{H,S}(\alpha, h),$$

$$M^*_{H,S}(\alpha) \equiv M_{H,S}(\alpha, T_{H,S}(\alpha)).$$

Because in general we may not know exact Lipschitz-constant for the Hessians, we do an adaptive search for estimating H. Resulting primal-dual scheme is presented in Algorithm 2. When a small subset of coordinates S is used, the most expensive operations become: computation of the objective at a current point $D(\alpha)$ and the matrix-vector product $\mathbf{B}^T \alpha$. Both of them can be significantly optimized by storing already computed values in memory and updating only changed information on every step.

Algorithm 2 Stochastic Dual Cubic Newton Ascent (SDCNA)

Parameters: sampling distribution $\hat{S}$
Initialization: choose initial $\alpha^0 \in Q$ and $H_0 > 0$
for $k = 0, 1, 2, \ldots$ **do**
  Make a primal update $w^k := \nabla g^* \left(\frac{1}{\lambda m} \mathbf{B}^T \alpha^k \right)$
  Sample $S_k \sim \hat{S}$
  While $M^*_{H_k, S_k}(\alpha^k) > -D(\alpha^k + T_{H_k, S_k}(\alpha^k))$ do
    $H_k := 1/2 \cdot H_k$
  Make a dual update $\alpha^{k+1} := \alpha^k + T_{H_k, S_k}(x^k)$
  Set $H_{k+1} := 2 \cdot H_k$
end for

8. Numerical Experiments

Synthetic We consider the following synthetic regression task: $\min_{x \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{A}x - b\|_2^2 + \sum_{i=1}^N \frac{c_i}{6} |x_i|^3$ with randomly generated parameters and for different N. On each problem of this type we run Algorithm 1 and evaluate total computational time until reaching 10^{-12} accuracy in function residual. Using middle-size blocks of coordinates on each step is the best choice in terms of total computational time, comparing it with small coordinate subsets and with full-coordinate method. This agrees with the provided complexity estimates for the method: an increase of the batch size speeds up convergence rate linearly, but slows down the cost of one iteration cubically. Therefore, the optimum size of the block is on a medium level.

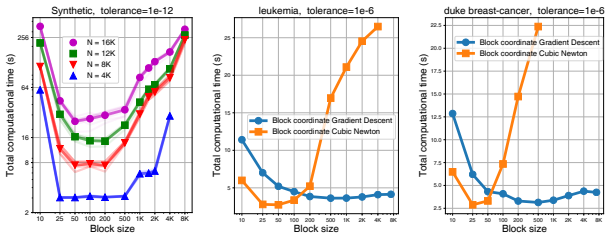

Figure 1. Time it takes to solve a problem for different sampling block sizes. Left: synthetic problem. Center and right: logistic regression for real data.

Logistic regression In this experiment we train ℓ_2-regularized logistic regression model for classification task with two classes by its constrained reformulation (16) and compare Algorithm 1 with the Block coordinate Gradient Descent on the datasets: *leukemia* ($m = 38, d = 7129$) and *duke breast-cancer* ($m = 44, d = 7129$). We see that using coordinate blocks of size $25 - 50$ for the Cubic Newton outperforms all other cases of both methods in terms of total computational time. Increasing block size further starts to significantly slow down the method because of high cost of every iteration.

Poisson regression In this experiment we train Poisson model for regression task with integer responses by the primal-dual Algorithm 2 and compare it with SDCA (Shalev-Shwartz & Zhang, 2013) and SDNA (Qu et al., 2016) methods on synthetic ($m = 1000, d = 200$) and real data ($m = 319, d = 20$). Our approach requires smaller number of epochs to reach given accuracy, but computational efficiency of every step is the same as in SDNA method.

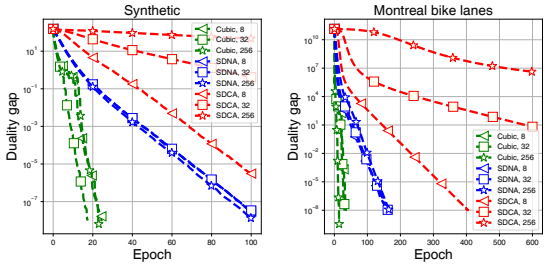

Figure 2. Comparison of Algorithm 2 (marked as Cubic) with SDNA and SDCA methods for minibatch sizes $\tau = 8, 32, 256$, training Poisson regression. Left: synthetic. Right: real data.

9. Acknowledgments

The work of the first author was partly supported by Samsung Research, Samsung Electronics, the Russian Science Foundation Grant 17-11-01027 and KAUST.

References

Agarwal, N., Allen-Zhu, Z., Bullins, B., Hazan, E., and Ma, T. Finding approximate local minima for nonconvex optimization in linear time. *arXiv preprint arXiv:1611.01146*, 2016.

Bertsekas, D. P. Local convex conjugacy and Fenchel duality. In *Preprints of 7th Triennial World Congress of IFAC, Helsinki, Finland*, volume 2, pp. 1079–1084, 1978.

Carmon, Y. and Duchi, J. C. Gradient descent efficiently finds the cubic-regularized non-convex newton step. *arXiv preprint arXiv:1612.00547*, 2016.

Cartis, C. and Scheinberg, K. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Mathematical Programming*, 169 (2):337–375, 2018.

Cartis, C., Gould, N. I., and Toint, P. L. Adaptive cubic regularisation methods for unconstrained optimization. part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011a.

Cartis, C., Gould, N. I., and Toint, P. L. Adaptive cubic regularisation methods for unconstrained optimization. part II: worst-case function-and derivative-evaluation complexity. *Mathematical programming*, 130(2):295–319, 2011b.

Conn, A. R., Gould, N. I., and Toint, P. L. *Trust region methods*. SIAM, 2000.

Ghadimi, S., Liu, H., and Zhang, T. Second-order methods with cubic regularization under inexact information. *arXiv preprint arXiv:1710.05782*, 2017.

Gould, N. I., Robinson, D. P., and Thorne, H. S. On solving trust-region and other regularised subproblems in optimization. *Mathematical Programming Computations*, 2 (1):21–57, 2010.

Grapiglia, G. N. and Nesterov, Y. Regularized Newton methods for minimizing functions with Hölder continuous Hessians. *SIAM Journal on Optimization*, 27(1): 478–506, 2017.

Griewank, A. The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical report, Technical Report NA/12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 1981.

Kohler, J. M. and Lucchi, A. Sub-sampled cubic regularization for non-convex optimization. *arXiv preprint arXiv:1705.05933*, 2017.

Mutnỳ, M. and Richtárik, P. Parallel stochastic newton method. *Journal of Computational Mathematics*, 36(3), 2018.

Nesterov, Y. Introductory lectures on convex optimization., 2004.

Nesterov, Y. Modified gauss–newton scheme with worst case guarantees for global performance. *Optimisation Methods and Software*, 22(3):469–483, 2007.

Nesterov, Y. Accelerating the cubic regularization of Newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.

Nesterov, Y. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

Nesterov, Y. and Polyak, B. T. Cubic regularization of Newton's method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Pilanci, M. and Wainwright, M. J. Randomized sketches of convex programs with sharp guarantees. *IEEE Transactions on Information Theory*, 61(9):5096–5115, 2015.

Qu, Z. and Richtárik, P. Coordinate descent with arbitrary sampling II: expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016.

Qu, Z., Richtárik, P., Takáč, M., and Fercoq, O. SDNA: stochastic dual Newton ascent for empirical risk minimization. In *International Conference on Machine Learning*, pp. 1823–1832, 2016.

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

Richtárik, P. and Takáč, M. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016.

Rockafellar, R. T. Convex analysis. Princeton landmarks in mathematics, 1997.

Shalev-Shwartz, S. and Zhang, T. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013.

Tripuraneni, N., Stern, M., Jin, C., Regier, J., and Jordan, M. I. Stochastic cubic regularization for fast nonconvex optimization. *arXiv preprint arXiv:1711.02838*, 2017.

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