Stochastic Subspace Cubic Newton Method

Filip Hanzely [1] **Nikita Doikov** [2] **Peter Richtárik** [1] **Yurii Nesterov** [2]

Abstract

In this paper, we propose a new randomized second-order optimization algorithm—Stochastic Subspace Cubic Newton (SSCN)—for minimizing a high dimensional convex function f. Our method can be seen both as a *stochastic* extension of the cubically-regularized Newton method of Nesterov and Polyak (2006), and a *second-order* enhancement of stochastic subspace descent of Kozak et al. (2019). We prove that as we vary the minibatch size, the global convergence rate of SSCN interpolates between the rate of stochastic coordinate descent (CD) and the rate of cubic regularized Newton, thus giving new insights into the connection between first and second-order methods. Remarkably, the local convergence rate of SSCN matches the rate of stochastic subspace descent applied to the problem of minimizing the quadratic function $\frac{1}{2}(x-x^*)^\top \nabla^2 f(x^*)(x-x^*)$, where x^* is the minimizer of f, and hence depends on the properties of f at the optimum only. Our numerical experiments show that SSCN outperforms non-accelerated first-order CD algorithms while being competitive to their accelerated variants.

1. Introduction

In this work we consider the optimization problem

$$\min_{x\in\mathbb{R}^d} \{F(x) := f(x) + \psi(x)\}\ , \tag{1}$$

where $f:\mathbb{R}^d\to\mathbb{R}$ is convex and twice differentiable and $\psi:\mathbb{R}^d\to\mathbb{R}\cup\{+\infty\}$ is a simple convex function. We are interested in the regime where the dimension d is very large, which arises in many contexts, such as the training of modern over-parameterized machine learning models. In this regime, coordinate descent (CD) methods, or more generally subspace descent methods, are the methods of choice.

1.1. Subspace descent methods

Subspace descent methods rely on update rules of the form

$$x^+ = x + \mathbf{S}h, \qquad \mathbf{S}\in\mathbb{R}^{d\times\tau(\mathbf{S})}, \qquad h\in\mathbb{R}^{\tau(\mathbf{S})}, \tag{2}$$

where $\mathbf{S}$ is a thin matrix, typically with a negligible number of columns compared to the dimension (i.e., $\tau(\mathbf{S})\ll d$). That is, they move from x to x^+ along the subspace spanned by the columns of $\mathbf{S}$.

In these methods, the subspace matrix $\mathbf{S}$ is typically chosen first, followed by the determination of the parameters h which define the linear combination of the columns determining the update direction. Several different rules have been proposed in the literature for choosing the matrix $\mathbf{S}$, including greedy, cyclic and randomized rules. In this work we consider a *randomized* rule. In particular, we assume that $\mathbf{S}$ is sampled from an arbitrary but fixed distribution $\mathcal{D}$ restricted to requiring that $\mathbf{S}$ be of full column rank[1] with probability one.

Once $\mathbf{S}\sim\mathcal{D}$ is sampled, a rule for deciding the stepsize h varies from algorithm to algorithm, but is mostly determined by the underlying *oracle model* for information access to function f. For instance, first-order methods require access to the subspace gradient $\nabla_{\mathbf{S}} f(x) := \mathbf{S}^\top \nabla f(x)$, and are relatively well studied (Nesterov, 2012; Stich et al., 2013; Richtárik & Takáč, 2014; Wright, 2015; Kozak et al., 2019). At the other extreme are variants performing a full subspace minimization, i.e., f is minimized over the affine subspace given by

$$\{x + \mathbf{S}h \,|\, h\in\mathbb{R}^{\tau(\mathbf{S})}\};$$

see (Chang et al., 2008). In particular, in this paper we are interested in the *second-order* oracle model; i.e., we

[1]King Abdullah University of Science and Technology, Thuwal, Saudi Arabia [2]Catholic University of Louvain, Louvain-la-Neuve, Belgium. Correspondence to: Filip Hanzely <fhanzely@gmail.com>, Nikita Doikov <nikita.doikov@uclouvain.be>, Peter Richtárik <peter.richtarik@kaust.edu.sa>, Yurii Nesterov <yurii.nesterov@uclouvain.be>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

[1]It is rather simple to extend our results to matrices $\mathbf{S}$ which are column-rank deficient. However, this would introduce a rather heavy notation burden which we decided to avoid for the sake of clarity and readability.

claim access both to the subspace gradient $\nabla_{\mathbf{S}} f(x)$ and the subspace Hessian $\nabla^2_{\mathbf{S}} f(x) := \mathbf{S}^\top \nabla^2 f(x) \mathbf{S}$.

1.2. Contributions

We now summarize our contributions:

(a) **New 2nd order subspace method.** We propose a new stochastic subspace method—Stochastic Subspace Cubic Newton (SSCN)—constructed by minimizing an oracle-consistent global upper bound on the objective f in each iteration (Section 3). This bound is formed using both the subspace gradient and the subspace Hessian at the current iterate and relies on Lipschitzness of the subspace Hessian.

(b) **Interpolating global rate.** We prove (Section 5) that SSCN enjoys a global convergence rate that interpolates between the rate of stochastic CD and the rate of cubic regularized Newton as one varies the expected dimension of the subspace, $\mathbb{E}[\tau(\mathbf{S})]$.

(c) **Fast local rate.** Remarkably, we establish a local convergence bound for SSCN (Section 6) that matches the rate of stochastic subspace descent (SSD) (Gower & Richtárik, 2015) applied to solving the problem

$$\min_{x \in \mathbb{R}^d} \frac{1}{2}(x - x^*)^\top \nabla^2 f(x^*)(x - x^*), \tag{3}$$

where x^* is the solution of (1). Thus, SSCN behaves as *if* it had access to a perfect second order model of f at the optimum, and was given the (intuitively much simpler) task of minimizing this model instead. Furthermore, note that SSD (Gower & Richtárik, 2015) applied to minimize a convex quadratic can be interpreted as doing an exact subspace search in each iteration, i.e., it minimizes the objective exactly along the active subspace (Richtárik & Takáč, 2017). Therefore, the local rate of SSCN matches the rate of the greediest strategy for choosing h in the active subspace, and as such, this rate is the best one can hope for a method that does not incorporate some form of acceleration.

(d) **Special cases.** We discuss in Section 3.2 how SSCN reduces to several existing stochastic second order methods in special cases, either recovering the best known rates, or improving upon them. This includes SDSA (Gower & Richtárik, 2015), CN (Griewank, 1981; Nesterov & Polyak, 2006) and RBCN Doikov & Richtárik (2018). However, our method is more general and hence allows for more applications.

We discuss more remotely related literature in Section 4. We now give a simple example of our setting.

Example 1 (Coordinate subspace setup). Let $\mathbf{I}^d \in \mathbb{R}^{d \times d}$ be the identity and let S be a random subset of $\{1, 2, \ldots, d\}$. Given that $\mathbf{S} = \mathbf{I}^d_{(:,S)}$ with probability 1, the oracle model reveals $(\nabla f(x))_S$ and $(\nabla^2 f(x))_{(S,S)}$. Therefore, we have access to a random block of partial derivatives of f and a block submatrix of its Hessian, both corresponding to the subset of indices S. Furthermore, the rule (2) updates a subset S of coordinates only. In this setting, our method is a new *second-order coordinate subspace descent* method.

2. Preliminaries

Throughout the paper, we assume that f is convex, twice differentiable, and sufficiently smooth and that ψ is convex, albeit possibly non-differentiable.[2]

Assumption 2.1. *Function $f : \mathbb{R}^d \to \mathbb{R}$ is convex and twice differentiable with M-Lipschitz continuous Hessian. Function $\psi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is proper closed and convex.*

We always assume that a minimum of F exists and by x^* denote any of its minimizers. We let $F^* := F(x^*)$.

Since our method always takes steps along random subspaces spanned by the columns of $\mathbf{S} \in \mathbb{R}^{d \times \tau(\mathbf{S})}$, it is reasonable to define the Lipschitzness of the Hessian over the range of $\mathbf{S}$:[3]

$$M_{\mathbf{S}} := \max_{x \in \mathbb{R}^d} \max_{\substack{h \in \mathbb{R}^{\tau(\mathbf{S})}, \\ h \neq 0}} \frac{|\nabla^3 f(x)[\mathbf{S}h]^3|}{\|\mathbf{S}h\|^3}. \tag{4}$$

As the next lemma shows, the maximal value of $M_{\mathbf{S}}$ for any $\mathbf{S}$ of width τ can be up to $(\frac{d}{\tau})^{\frac{3}{2}}$ times smaller than M and this will lead to a tighter approximation of the objective.

Lemma 2.2. *We have*

$$M \geq \max_{\tau(\mathbf{S}) = \tau} M_{\mathbf{S}}.$$

Moreover, there is a problem where

$$\max_{\tau(\mathbf{S}) = \tau} M_{\mathbf{S}} = \left(\frac{\tau}{d}\right)^{\frac{3}{2}} M.$$

Lastly, if $\mathrm{Range}(\mathbf{S}) = \mathrm{Range}(\mathbf{S}')$, *then* $M_{\mathbf{S}} = M_{\mathbf{S}'}$.

The next lemma provides a direct motivation for our algorithm. It gives a global upper bound on the objective over a random subspace, given the first and second-order information at the current point.

Lemma 2.3. *Let $x \in \mathbb{R}^d$, $\mathbf{S} \in \mathbb{R}^{d \times \tau(\mathbf{S})}$, $h \in \mathbb{R}^{\tau(\mathbf{S})}$ and x^+ be as in (2). Then*

$$\begin{aligned} |f(x^+) - f(x) - \langle \nabla_{\mathbf{S}} f(x), h \rangle - \tfrac{1}{2} \langle \nabla^2_{\mathbf{S}} f(x) h, h \rangle| \\ \leq \tfrac{M_{\mathbf{S}}}{6} \|\mathbf{S}h\|^3. \end{aligned} \tag{5}$$

[2] We will also require separability of ψ; see Section 5.1.

[3] By $\|x\| := \langle x, x \rangle^{1/2}$ we denote the standard Euclidean norm.

As a consequence, we have

$$F(x^+) \leq f(x) + T_{\mathbf{S}}(x,h), \tag{6}$$

where $T_{\mathbf{S}}(x,h) := \langle \nabla_{\mathbf{S}} f(x), h\rangle + \frac{1}{2}\langle \nabla^2_{\mathbf{S}} f(x)h, h\rangle + \frac{M_{\mathbf{S}}}{6}\|\mathbf{S}h\|^3 + \psi(x+\mathbf{S}h)$.

We shall also note that for function ψ we require *separability* with respect to the sampling distribution (see Definition 5.5 and the corresponding Assumption 5.6 in Section 5.1).

For better orientation throughout the paper, we provide a table of frequently used notation in the Appendix.

3. Algorithm

For a given $\mathbf{S}$ and current iterate x^k, it is a natural idea to choose h as a minimizer of the upper bound (6) in h for $x = x^k$, and subsequently set $x^{k+1} = x^+$ via (2). Note that we are choosing $\mathbf{S}$ randomly according to a fixed distribution $\mathcal{D}$ (with a possibly random number of columns). We have just described SSCN—Stochastic Subspace Cubic Newton—formally stated as Algorithm 1.

Algorithm 1 SSCN: Stochastic Subspace Cubic Newton

1: **Initialization:** x^0, distribution $\mathcal{D}$ of random matrices with d rows and full column rank
2: **for** $k = 0, 1, \ldots$ **do**
3: Sample $\mathbf{S}$ from distribution $\mathcal{D}$
4: $h^k = \operatorname{argmin}_{h\in\mathbb{R}^{\tau(\mathbf{S})}} T_{\mathbf{S}}(x^k, h)$
5: Set $x^{k+1} = x^k + \mathbf{S}h^k$
6: **end for**

Remark 1. Inequality (6) becomes an equality with $h = 0$. As a consequence, we must have $F(x^{k+1}) \leq F(x^k)$, and thus the sequence $\{F(x^k)\}_{k\geq 0}$ is non-increasing.

3.1. Solving the subproblem

Algorithm 1 requires $T_{\mathbf{S}}$ to be minimized in h each iteration. As this operation does not have a closed-form solution in general, it requires an optimization subroutine itself of a possibly non-trivial complexity, which we discuss here.

The subproblem without ψ. Let us now consider the case when $\psi(x) \equiv 0$ in which our problem (1) does not contain any nondifferentiable components. Various techniques for minimizing regularized quadratic functions were developed during the development of Trust-region methods (see (Conn et al., 2000)), and applied to Cubic regularization in (Nesterov & Polyak, 2006). The classical approach consists in performing some diagonalization of the matrix $\nabla^2_{\mathbf{S}} f(x)$ first, by computing the *eigenvalue* or *tridiagonal* decomposition, which costs $\mathcal{O}(\tau(\mathbf{S})^3)$ arithmetical operations. Then, to find the minimizer, it merely remains to solve a one-dimensional nonlinear equation (this part can be done by $\tilde{\mathcal{O}}(1)$ iterations of the one-dimensional Newton method, with a linear cost per step). More details and analysis of this procedure can be found in (Gould et al., 2010).

The next example gives a setting in which an explicit formula for the minimizer of $T_{\mathbf{S}}$ can be deduced.

Example 2. Let e_i be the ith unit basis vector in $\mathbb{R}^d$. If $\mathbf{S} \in \{e_1, \ldots, e_d\}$ with probability 1 and $\psi(x) = 0$, the update rule can be written as $x^{k+1} = x^k - \alpha_i^k e_i$, with

$$\alpha_i^k = \frac{2\nabla_i f(x^k)}{\nabla^2_i f(x^k) + \sqrt{\left(\nabla^2_{ii} f(x^k)\right)^2 + 2M_{e_i}|\nabla_i f(x^k)|}},$$

thus the cost of solving the subproblem is $\mathcal{O}(1)$.

Subproblem with simple ψ. In some scenarios, minimization of $T_{\mathbf{S}}$ can be done using a simple algorithm if ψ is simple enough. We now give an example of this.

Example 3. If $\mathbf{S} \in \{e_1, \ldots, e_d\}$ with probability 1, the subproblem can be solved using a binary search given that the evaluation of ψ is cheap. In particular, if we can evaluate $\psi(x^k + \mathbf{S}h) - \psi(x^k)$ in $\tilde{\mathcal{O}}(1)$, the cost of solving the subproblem will be $\tilde{\mathcal{O}}(1)$.

The subproblem with general ψ. In the case of general regularizers, recent line of work by Carmon & Duchi (2019) explores to the use of *first-order* optimization methods (Gradient Methods) for computing an approximate minimizer of $T_{\mathbf{S}}$. We note that the backbone of such Gradient Methods is an implementation of the following operation (for a given vector $b \in \mathbb{R}^{\tau(\mathbf{S})}$, and positive scalars α, β):

$$\arg\min_{h\in\mathbb{R}^{\tau(\mathbf{S})}} \langle b, h\rangle + \frac{\alpha}{2}\|\mathbf{S}h\|^2 + \frac{\beta}{3}\|\mathbf{S}h\|^3 + \psi(x^k + \mathbf{S}h).$$

To the best of our knowledge, the most efficient gradient method is the Fast Gradient Method (FGM) of Nesterov (2019), achieving an $\mathcal{O}(1/k^6)$ convergence rate. However, FGM can deal with any ψ as long as the above subproblem is cheap to solve. We shall also note that gradient methods do not require a storage of $\nabla^2_{\mathbf{S}} f(x)$; but rather iteratively access partial Hessian-vector products $\nabla^2_{\mathbf{S}} f(x)h$.

Line search. Note that in Algorithm 1 we use the Lipschitz constants $M_{\mathbf{S}}$ of the subspace Hessian (see Definition (4)) as the regularization parameters. In many application, $M_{\mathbf{S}}$ can be estimated cheaply (see Section 7). In general, however, $M_{\mathbf{S}}$ might be unknown or hard to estimate. In such a case, one might use a simple one-dimensional search on each iteration: multiply the estimate of $M_{\mathbf{S}}$ by the factor of two until the bound (6) is satisfied, and divide it by two at the start of each iteration. Note that the average number of such line search steps per iteration can be bounded by two (see (Grapiglia & Nesterov, 2017) for the details).

3.2. Special cases

There are several scenarios where SSCN becomes an already known algorithm. We list them below:

- **Quadratic minimization.** If $M = 0$ and $\psi = 0$, SSCN reduces to the stochastic dual subspace ascent (SDSA) method (Gower & Richtárik, 2015), first analyzed in an equivalent primal form as a *sketch-and-project* method in (Gower & Richtárik, 2015). In such a case, SSCN performs both first-order, second-order updates, and exact minimization over a subspace at the same time due to the quadratic structure of the objective (Richtárik & Takáč, 2017). The convergence rate we provide in Section 6 exactly matches the rate of sketch-and-project as well. As a consequence, we recover a subclass of matrix inversion algorithms (Gower & Richtárik, 2017) together with stochastic spectral (coordinate) descent (Kovalev et al., 2018) along with their convergence theory.
- **Full-space method.** If $\mathbf{S} = \mathbf{I}^d$ with probability 1, SSCN reduces to cubically regularized Newton (CN) (Griewank, 1981; Nesterov & Polyak, 2006). In this case, we recover both existing global convergence rates and superlinear local convergence rates.
- **Separable non-quadratic part of f.** The RBCN method of Doikov & Richtárik (2018) aims to minimize (1) with

  $$f(x) = g(x) + \phi(x),$$

  where g, ϕ are both convex, and ϕ is separable.[4] They assume that

  $$\nabla^2 g(x) \preceq \mathbf{A} \in \mathbb{R}^{d\times d}, \qquad \forall x \in \mathbb{R}^d,$$

  while ϕ has Lipschitz continuous Hessian. In each iteration, RBCN constructs an upper bound on the objective using first order information from g only. This is unlike SSCN, which uses second order information from g. In a special case when $\nabla^2 g(x) = \mathbf{A}$ for all x, SSCN and RBCN are identical algorithms. However, RBCN is less general: it requires separable ϕ, and thus does not cover some of our applications, and takes directions along coordinates only. Further, the rates we provide are better even in the setting where the two methods coincide ($\nabla^2 g(x) = \mathbf{A}$). The simplest way to see that is by looking at local convergence – RBCN does not achieve the local convergence rate of block CD to minimize (3), which is the best one might hope for.

Besides these particular cases, for a general twice-differentiable f, SSCN is a new second-order method.

[4] Separability is defined in Section 5.1.

4. Related Literature

Several methods in the literature are related to SSCN. We briefly review them below.

- *Cubic regularization of Newton method* was proposed first by Griewank (1981), and received substantial attention after the work of Nesterov & Polyak (2006), where its global complexity guarantees were established. During the last decade, there was a steady increase of research in second-order methods, discovering Accelerated (Nesterov, 2008; Monteiro & Svaiter, 2013), Adaptive (Cartis et al., 2011a;b), and Universal (Grapiglia & Nesterov, 2017; 2019; Doikov & Nesterov, 2019) schemes (the latter ones are adjusting automatically to the smoothness properties of the objective).
- There is a vast literature on *first-order coordinate descent (CD)* methods. While CD with $\tau = 1$ is consistently the same method within the literature (Nesterov, 2012; Richtárik & Takáč, 2014; Wright, 2015), there are several ways to deal with $\tau > 1$. The first approach constructs a separable upper bound on the objective (in expectation) in the direction of a random subset of coordinates (Qu & Richtárik, 2016a;b), which is minimized to obtain the next iterate. The second approach—SDNA (Qu et al., 2016)—works with a tighter non-separable upper bound. SDNA is, therefore, more costly to implement but requires a smaller number of iterations to converge. The literature on first-order subspace descent algorithms is slightly less rich, the notable examples are random pursuit (Stich et al., 2013) or stochastic subspace descent (Kozak et al., 2019).
- *Randomized subspace Newton* (RSN) (Gower et al., 2019) is a method of the form

  $$x^{k+1} = x^k - \frac{1}{\hat{L}}\mathbf{S}\left(\nabla^2_{\mathbf{S}} f(x^k)\right)^{-1} \nabla_{\mathbf{S}} f(x^k)$$

  for some specific fixed $\hat{L}$. In particular, it can be seen as a method minimizing the following upper bound on the function, which follows from their assumption:

  $$h^k = \arg\min_h \langle \nabla_{\mathbf{S}} f(x^k), h\rangle + \frac{\hat{L}}{2}\langle \nabla^2_{\mathbf{S}} f(x^k)h, h\rangle.$$

  This is followed by an update over the subspace: $x^{k+1} = x^k + \mathbf{S}h^k$. Since both RSN and SSCN are analyzed under different assumptions, the global linear rates are not directly comparable. However, the local rate of SSCN is superior to RSN. We shall also note that RSN is a stochastic subspace version of a method from (Karimireddy et al., 2018).

- *Subsampled Newton* (SN) methods (Byrd et al., 2011; Erdogdu & Montanari, 2015; Xu et al., 2016; Roosta-Khorasani & Mahoney, 2019) and *subsampled cubic regularized Newton methods* (Kohler & Lucchi, 2017; Xu et al., 2017; Wang et al., 2018) and *stochastic (cubic regularized) Newton methods* (Tripuraneni et al., 2018; Cartis & Scheinberg, 2018; Kovalev et al., 2019) are stochastic second-order algorithms to tackle finite sum minimization. Their major disadvantage is a requirement of an immense sample size, which makes them often impractical if used as theory prescribes. A notable exception that does not require a large sample size was recently proposed by Kovalev et al. (2019). However, none of these methods are directly comparable to SSCN as they are not subspace descent methods, but rather randomize over data points (or sketch the Hessian from "inside" (Pilanci & Wainwright, 2017)).

5. Global Complexity Bounds

We first start presenting the global complexity results of SSCN.

5.1. Setup

Throughout this section, we require some kind of uniformity of the distribution $\mathcal{D}$ over subspaces given by $\mathbf{S}$. In particular, we require

$$\mathbf{P}^{\mathbf{S}} := \mathbf{S}\left(\mathbf{S}^\top \mathbf{S}\right)^{-1}\mathbf{S}^\top,$$

the projection matrix onto the range of $\mathbf{S}$, to be a scalar multiple of identity matrix in expectation.

Assumption 5.1. *$\exists \tau > 0$ such that distribution $\mathcal{D}$ satisfies*

$$\mathbb{E}\left[\mathbf{P}^{\mathbf{S}}\right] = \frac{\tau}{d}\mathbf{I}^d. \tag{7}$$

A direct consequence of Assumption 5.1 is that τ is an expected width of $\mathbf{S}$, as the next lemma states.

Lemma 5.2. *If Assumption 5.1 holds, then $\mathbb{E}\left[\tau(\mathbf{S})\right] = \tau$.*

As mentioned before, the global complexity results are interpolating between convergence rate of (first-order) CD and (global) convergence rate of Cubic Newton. However, first-order CD requires Lipschitzness of gradients, and thus we will require it as well.

Assumption 5.3. *Function f has L-Lipschitz continuous gradients, i.e., $\nabla^2 f(x) \preceq L\mathbf{I}^d$ for all $x \in \mathbb{R}^d$.*

We will also need an extra assumption on ψ. It is well known that proximal (first-order) CD with fixed step size does not converge if ψ is not separable – in such case, even if $f(x^k) = f(x^*)$ we might have $f(x^{k+1}) > f(x^*)$. Therefore, we might not hope that SSCN will converge without additional assumptions on ψ. Informally speaking, separability of ψ with respect to directions given by columns of $\mathbf{S}$ is required. To define it formally, let us introduce first the notion of a separable set.

Definition 5.4. *Set $Q \subseteq \mathbb{R}^d$ is called D-separable, if $\forall x, y \in Q, \mathbf{S} \in D$: $\mathbf{P}^{\mathbf{S}}x + (\mathbf{I}^d - \mathbf{P}^{\mathbf{S}})y \in Q$.*

Using the set separability, we next define a separability of a function.

Definition 5.5. *Function $\phi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is D-separable if $\operatorname{dom}\phi$ is D-separable, and there is map $\phi' : \operatorname{dom}\phi \to \mathbb{R}^d$ such that*

1. $\forall x \in \operatorname{dom}\phi:\ \phi(x) = \langle \phi'(x), e\rangle$,[5]
2. $\forall x, y \in \operatorname{dom}\phi, \mathbf{S} \in D:\ \phi'(\mathbf{P}^{\mathbf{S}}x + (\mathbf{I}^d - \mathbf{P}^{\mathbf{S}})y) = \mathbf{P}^{\mathbf{S}}\phi'(x) + (\mathbf{I}^d - \mathbf{P}^{\mathbf{S}})\phi'(y)$.

Example 4. If D is a set of matrices whose columns are standard basis vectors, D-separability reduces to classical (coordinate-wise) separability.

Example 5. If D is set of matrices which are column-wise submatrices of orthogonal $\mathbf{U}$, D-separability of ϕ reduces to classical coordinate-wise separability of $\phi(\mathbf{U}^\top x)$.

Example 6. $\phi(x) = \frac{1}{2}\|x\|^2$ is D-separable for any D.

Assumption 5.6. *Function ψ is $\operatorname{Range}(\mathcal{D})$-separable.*

We are now ready to present the convergence rate of SSCN.

5.2. Theory

First, let us introduce the critical lemma from which the main global complexity results are derived. The next lemma states, what is the expected progress we have for one step of SSCN.

Lemma 5.7. *Let Assumptions 2.1, 5.1, 5.3 and 5.6 hold. Then, for every $k \geq 0$ and $y \in \mathbb{R}^d$ we have*

$$\begin{aligned}\mathbb{E}\left[F(x^{k+1}) \,|\, x^k\right] \;\leq\; &\left(1 - \tfrac{\tau}{d}\right)F(x^k) + \tfrac{\tau}{d}F(y) \\ &+ \tfrac{\tau}{d}\left(\tfrac{d-\tau}{d}\tfrac{L}{2}\|y - x^k\|^2 + \tfrac{M}{3}\|y - x^k\|^3\right).\end{aligned} \tag{8}$$

Now we are ready to present global complexity results for the general class of convex functions. The convergence rate is obtained by summing (8) over the different iterations k, and with a specific choice of y.

Theorem 5.8. *Let Assumptions 2.1, 5.1, 5.3 and 5.6 hold. Denote*

$$R \overset{\text{def}}{=} \sup_{x \in \mathbb{R}^d}\left\{\|x - x^*\| \;:\; F(x) \leq F(x^0)\right\}, \tag{9}$$

[5] By $e \in \mathbb{R}^d$ we mean the vector of all ones.

and suppose that $R < +\infty$. Then, for every $k \geq 1$ we have

$$\mathbb{E}\left[F(x^k)\right] - F^* \leq \frac{d-\tau}{\tau} \cdot \frac{4.5LR^2}{k} + \left(\frac{d}{\tau}\right)^2 \cdot \frac{9MR^3}{k^2} + \frac{F(x^0)-F^*}{1+\frac{1}{4}\left(\frac{\tau}{d}k\right)^3}. \quad (10)$$

Note that convergence rate of the minibatch version[6] of first-order CD is $\mathcal{O}\left(\frac{d}{\tau}\frac{LR^2}{k}\right)$. At the same time, (global) convergence rate of cubically regularized Newton method is $\mathcal{O}\left(\frac{MR^3}{k^2}\right)$. Therefore, Theorem 5.8 shows that the global rate of SSCN well interpolates between the two extremes, depending on the sample size τ we choose.

Remark 2. According to estimate (10), in order to have $\mathbb{E}\left[F(x^k)\right] - F^* \leq \varepsilon$, it is enough to perform

$$k = \mathcal{O}\left(\frac{d-\tau}{\tau}\frac{LR^2}{\varepsilon} + \frac{d}{\tau}\sqrt{\frac{MR^3}{\varepsilon}} + \frac{d}{\tau}\left(\frac{F(x^0)-F^*}{\varepsilon}\right)^{1/3}\right)$$

iterations of SSCN.

Next, we move to the strongly convex case.

Assumption 5.9. *Function f is μ-strongly convex, i.e., $\nabla^2 f(x) \succeq \mu \mathbf{I}^d$ for all $x \in \mathbb{R}^d$.*

Remark 3. Strong convexity of the objective (assumed for Theorem 5.10 later) implies: $R < +\infty$. Furthermore, due to monotonicity of the sequence $\{F(x_k)\}_{k\geq 0}$ (see Remark 1), we have $\|x^k - x^*\| \leq R$ for all k. Therefore, it is sufficient to require Lipschitzness of gradients over the sublevel set, which holds with $L = \lambda_{\max}(\nabla^2 f(x^*)) + MR$.

As both extremes cubic regularized Newton (where $\mathbf{S} = \mathbf{I}^d$ always) and (first-order) CD ($\mathbf{S} = e_i$ for randomly chosen i) enjoy (global) linear rate under strong convexity, linear convergence of SSCN is expected as well. At the same time, the leading complexity term should be in between the two extremes. Such a result is established as Theorem 5.10.

Theorem 5.10. *Let Assumptions 2.1, 5.1, 5.6 and 5.9 hold. Then, $\mathbb{E}\left[F(x^k)\right] - F^* \leq \varepsilon$, as long as the number of iterations of SSCN is*

$$k = \mathcal{O}\left(\left(\frac{d-\tau}{\tau}\frac{L}{\mu} + \frac{d}{\tau}\sqrt{\frac{MR}{\mu}} + \frac{d}{\tau}\right) \cdot \log\frac{F(x^0)-F^*}{\varepsilon}\right).$$

Indeed, if $\mathbf{S} = \mathbf{I}^d$ with probability 1 and $MR \geq \mu$, the leading complexity term becomes $\sqrt{\frac{MR}{\mu}}\log\frac{1}{\varepsilon}$ which corresponds to the global complexity of cubically regularized Newton for minimizing strongly convex functions (Nesterov & Polyak, 2006). On the other side of the spectrum if $\mathbf{S} = e_i$ with probability $\frac{1}{d}$, the leading complexity term becomes $\frac{dL}{\mu}\log\frac{1}{\varepsilon}$, which again corresponds to convergence rate of CD (Nesterov, 2012). Lastly, if $1 < \tau < d$, the global linear rate interpolates the rates mentioned above.

[6] Sampling τ coordinates at a time for objectives with L-Lipschitz gradients.

Remark 4. Proof of Theorem 5.10 only uses the following consequence of strong convexity:

$$\frac{\mu}{2}\|x - x^*\|^2 \leq F(x) - F^*, \qquad x \in \mathbb{R}^d \quad (11)$$

and thus the conditions of Theorem 5.10 might be slightly relaxed.[7] For detailed comparison of various relaxations of strong convexity, see (Karimi et al., 2016).

6. Local Convergence

Throughout this section, assume that $\psi = 0$. We first present the key descent lemma, which will be used to obtain local rates. Let

$$\mathbf{H}_{\mathbf{S}}(x) := \nabla^2_{\mathbf{S}} f(x) + \sqrt{\frac{M_{\mathbf{S}}}{2}}\|\nabla_{\mathbf{S}} f(x)\|^{\frac{1}{2}}\mathbf{I}^{\tau(\mathbf{S})}.$$

Lemma 6.1. *We have*

$$f(x^k) - f(x^{k+1}) \geq \frac{1}{2}\|\nabla_{\mathbf{S}} f(x^k)\|^2_{\mathbf{H}_{\mathbf{S}}^{-1}(x^k)}. \quad (12)$$

Before stating the convergence theorem, it will be suitable to define the stochastic condition number of $\mathbf{H}_* := \nabla^2 f(x^*)$:

$$\zeta := \lambda_{\min}\left(\mathbf{H}_*^{\frac{1}{2}}\mathbb{E}\left[\mathbf{S}\left(\mathbf{S}^\top\mathbf{H}_*\mathbf{S}\right)^{-1}\mathbf{S}^\top\right]\mathbf{H}_*^{\frac{1}{2}}\right), \quad (13)$$

as it will drive the local convergence rate of SSCN.

Theorem 6.2 (Local Convergence)**.** *Let Assumptions 2.1, 5.9 hold, and suppose that $\psi = 0$. For any $\epsilon > 0$ there exists $\delta > 0$ such that if $F(x^0) - F^* \leq \delta$, we have*

$$\mathbb{E}\left[F(x^k) - F^*\right] \leq (1 - (1-\epsilon)\zeta)^k\left(F(x^0) - F^*\right) \quad (14)$$

and therefore the local complexity of SSCN is

$$\mathcal{O}\left(\zeta^{-1}\log\frac{1}{\varepsilon}\right).$$

If further $M = 0$ (i.e., f is quadratic), then $\epsilon = 0$ and $\delta = \infty$, and thus the rate is global.

The proof of Theorem 6.2 along with the exact formulas for ε, δ can be found in Section E of the Appendix. Theorem 6.2 provides a local linear convergence rate of SSCN. While one might expect a superlinear rate to be achievable, this is not the case, and we argue that the rate from Theorem 6.2 is the best one can hope for.

In particular, if $M = 0$, Algorithm 1 becomes subspace descent for minimizing positive definite quadratic which is a specific instance of sketch-and-project (Gower & Richtárik,

[7] However, this relaxation is not sufficient to obtain the local convergence results.

2015). However, sketch-and-project only converges linearly – the iteration complexity of sketch-and-project to minimize $(x-x^*)^\top \mathbf{A}(x-x^*)$ with $\mathbf{A} \succ 0$ is

$$\mathcal{O}\left(\left[\lambda_{\min}\left(\mathbf{A}^{\frac{1}{2}}\mathbb{E}\left[\mathbf{S}\left(\mathbf{S}^\top \mathbf{A}\mathbf{S}\right)^{-1}\mathbf{S}^\top\right]\mathbf{A}^{\frac{1}{2}}\right)\right]^{-1}\log\frac{1}{\varepsilon}\right).$$

Notice that this rate is matched by Theorem 6.2 in this case.

Next, we compare the local rate of SSCN to the rate of SDNA (Qu et al., 2016). To best of our knowledge, SDNA requires the least oracle calls to minimize f among all first-order non-accelerated methods.

Remark 5. SDNA is a first-order analogue to Algorithm 1 with $\mathbf{S} = \mathbf{I}^d_{(:,S)}$. In particular, given matrix $\mathbf{L}$ such that $\mathbf{L} \succeq \nabla^2 f(x) \succ 0$ for all x, the update rule of SDNA is

$$x^+ = x - \mathbf{S}\left(\mathbf{S}^\top \mathbf{L}\mathbf{S}\right)^{-1}\nabla_{\mathbf{S}} f(x),$$

where $\mathbf{S} = \mathbf{I}^d_{(:,S)}$ for a random subset of columns S. SDNA enjoys linear convergence rate with leading complexity term $\left(\mu\lambda_{\min}\left(\mathbb{E}\left[\mathbf{S}(\mathbf{S}^\top\mathbf{L}\mathbf{S})^{-1}\mathbf{S}^\top\right]\right)\right)^{-1}$. The leading complexity term of SSCN is ζ^{-1}, and we can bound

$$\begin{aligned}\zeta &\geq \lambda_{\min}(\mathbf{H}_*)\,\lambda_{\min}\left(\mathbb{E}\left[\mathbf{S}\left(\mathbf{S}^\top\mathbf{H}_*\mathbf{S}\right)^{-1}\mathbf{S}^\top\right]\right)\\ &\geq \mu\lambda_{\min}\left(\mathbb{E}\left[\mathbf{S}\left(\mathbf{S}^\top\mathbf{L}\mathbf{S}\right)^{-1}\mathbf{S}^\top\right]\right).\end{aligned}$$

Hence, the local rate of SSCN is no worse than the rate of SDNA. Furthermore, both of the above inequalities might be very loose in some cases (i.e., there are examples where $\frac{\zeta}{\mu\lambda_{\min}\mathbb{E}[\mathbf{S}(\mathbf{L}\mathbf{S})^{-1}\mathbf{S}^\top]}$ can be arbitrarily high). Therefore, local convergence rate of SSCN might be arbitrarily better than the convergence rate of SDNA. As a consequence, the local convergence of SSCN is better than convergence rate of any non-accelerated first order method.[8].

Lastly, the local convergence rate provided by Theorem 6.2 recovers the superlinear rate of cubic regularized Newton's method, as the next remark states.

Remark 6. If $\mathbf{S} = \mathbf{I}^d$ with probability 1, Algorithm 1 becomes cubic regularized Newton method (Griewank, 1981; Nesterov & Polyak, 2006). For $\mathbf{H}_* := \nabla^2 f(x^*)$ we have

$$\zeta = \lambda_{\min}\left(\mathbf{H}_*^{\frac{1}{2}}\mathbf{H}_*^{-1}\mathbf{H}_*^{\frac{1}{2}}\right) = \lambda_{\min}(\mathbf{I}^d) = 1.$$

As a consequence of Theorem 6.2, for any $\varepsilon > 0$ there exists $\delta > 0$ such that if $F(x) - F(x^*) \leq \delta$, we have

$$F(x^+) - F(x^*) \leq \varepsilon(F(x) - F(x^*)).$$

Therefore, we obtain a superlinear convergence rate.

[8]The rate of SSCN and rate of accelerated subspace descent methods are not directly comparable – while the (local) rate of SSCN might be better than rate of ACD, the reverse might happen as well. However, both ACD and SSCN are faster than non-accelerated subspace descent.

7. Applications

7.1. Linear Models

Consider only $\mathbf{S} = \mathbf{I}^d_{(:,S)}$ for simplicity. Let

$$F(x) := \frac{1}{n}\sum_{i=1}^n \phi_i(\langle a_i, x\rangle) + \psi(x), \tag{15}$$

and $f(x) := \frac{1}{n}\sum_{i=1}^n \phi_i(\langle a_i, x\rangle)$ and suppose that $|\nabla^3\phi_i(y)| \leq c$. Then clearly,

$$\nabla^3 f(x)[h]^3 = \frac{1}{n}\sum_{i=1}^n \nabla^3\phi_i(\langle a_i, x\rangle)\langle a_i, h\rangle^3$$

for any $h \in \mathbb{R}^d$. While evaluating $E := \max_{\|h\|=1,x}\nabla^3 f(x)[h]^3$ is infeasible, we might bound it instead via

$$E \leq \max_{\|h\|=1}\frac{c}{n}\sum_{i=1}^n |\langle a_i, h\rangle|^3 \leq \frac{c}{n}\sum_{i=1}^n \|a_i\|^3, \tag{16}$$

which means that $M = \frac{c}{n}\sum_{i=1}^n \|a_i\|^3$ is a feasible choice. On the other hand, for $S = \{j\}$ we have

$$\max_{\|h_j\|=1,x}\nabla^3 f(x)[h_j]^3 = \max_x \nabla^3 f(x)[e_j]^3 \leq \frac{c}{n}\sum_{i=1}^n |a_{ij}|^3$$

and thus we might set $M_j = \frac{c}{n}\sum_{i=1}^n |a_{ij}|^3$. The next lemma compares the above choices of M and M_j.

Lemma 7.1. *We have $M \geq \max_j M_j$. At the same time, there exist vectors $\{a_i\}$ that*

$$\max_j M_j = \frac{M}{d^{\frac{3}{2}}}.$$

Proof. The first part is trivial. For the second part, consider $a_{i,j} \in \{-1, 1\}$. □

Remark 7. One might avoid the last inequality from (16) using polynomial optimization; however, this might be more expensive than solving the original optimization problem and thus is not preferable. Another strategy would be to use a line search, see Section 3.1.

Both the formula for M and the formula for M_j require the prior knowledge of $c \geq 0$ such that $|\nabla^3\phi_i(y)| \leq c$ for all i. The next lemma shows how to compute such c for the logistic regression (binary classification model).

Lemma 7.2. *Let $\phi_i(y) = \log(1 + e^{-b_i y})$, where $b_i \in \{-1, 1\}$. Then $c = \frac{1}{6\sqrt{3}}$.*

Proof. $\nabla^3\phi_i(y) = -\frac{e^x(e^x-1)}{(1+e^x)^3} \Rightarrow \left|\nabla^3\phi_i(y)\right| \leq \frac{1}{6\sqrt{3}}$. □

Cost of performing a single iteration For the sake of simplicity, let $\tau(\mathbf{S}) = 1$, $\psi = 0$. Any CD method (i.e,. method with update rule (2) with $\mathbf{S} \in \{e_1, \dots, e_d\}$) can be efficiently implemented by memorizing the residuals $\langle a_i, x^k \rangle$, which is cheap to track since $x^{k+1} - x^k$ is a sparse vector. The overall cost of updating the residuals is $\mathcal{O}(n)$ while the cost of computing $\nabla_i f(x)$ and $\nabla^2_{i,i} f(x)$ (given the residuals are stored) is $\mathcal{O}(n)$. Therefore the overall cost of performing a single iteration is $\mathcal{O}(n)$. Generalizing to $\tau(\mathbf{S}) = \tau \geq 1$, the overall cost of single iteration of SSCN can be estimated as $\mathcal{O}(n\tau^2 + \tau^3)$, where $\mathcal{O}(n\tau^2)$ comes from evaluating subspace gradient and Hessian, while $\mathcal{O}(\tau^3)$ comes from solving the cubic subproblem.

7.2. Dual of linear models

So far, all results and applications for CRDS we mentioned were problems with large model size d. In this section we describe how SSCN can be efficient to tackle big data problems in some settings. Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ is data matrix and consider a specific instance of (15) where

$$\min_{x \in \mathbb{R}^d} F_P(x) := \frac{1}{d} \sum_{i=1}^{n} \rho_i(\mathbf{A}_{(:,i)} x) + \frac{\lambda}{2} \|x\|^2. \qquad (17)$$

where ρ_i is convex for all i. One can now formulate a dual problem of (17) as follows:

$$\max_{y \in \mathbb{R}^n} F_D(y) := -\frac{1}{2\lambda n^2} \left\| \mathbf{A}^\top y \right\|^2 - \frac{1}{n} \sum_{i=1}^{n} \rho_i^*(e_i^\top x). \quad (18)$$

Note that (18) is of form (15), and therefore if ρ_i^* has Lipschitz Hessian, we can apply SSCN to efficiently solve it (same as Section 7.1). Given the solution of (18), we can recover the solution of (17) (duality theory). Thus, SSCN can be used as a data-stochastic method to solve finite-sum optimization problems.

The trick described in this section is rather well known. It was first used in (Shalev-Shwartz & Zhang, 2013), where CD applied to the problem (18) (SDCA) was shown to be competitive with the variance reduced methods like SAG (Roux et al., 2012), SVRG (Johnson & Zhang, 2013) or SAGA (Defazio et al., 2014).

8. Experiments

We now numerically verify our theoretical claims. Due to space limitation, we only present a fraction of all experiments here, the remaining part, together with the exact setup for this experiment can be found in Section B of the Appendix.

We consider binary classification with LIBSVM (Chang & Lin, 2011) data modelled by regularized logistic regression. We compare SSCN against three different instances of (first-order) randomized coordinate descent: CD with uniform sampling, CD with importance sampling (Nesterov, 2012), and accelerated CD with importance sampling (Allen-Zhu et al., 2016; Nesterov & Stich, 2017).

In order to be comparable with the mentioned first-order methods, we consider $\mathbf{S} \in \{e_1, \dots, e_d\}$ with probability 1 – the complexity of performing each iteration is about the same for each algorithm now. At the same time, computing M_{e_i} for all $1 \leq i \leq d$ is of cost $\mathcal{O}(nd)$ – the same cost as computing coordinate-wise smoothness constants for (accelerated) coordinate descent (see Section 7.1 for the details). Figure 3 shows the result.

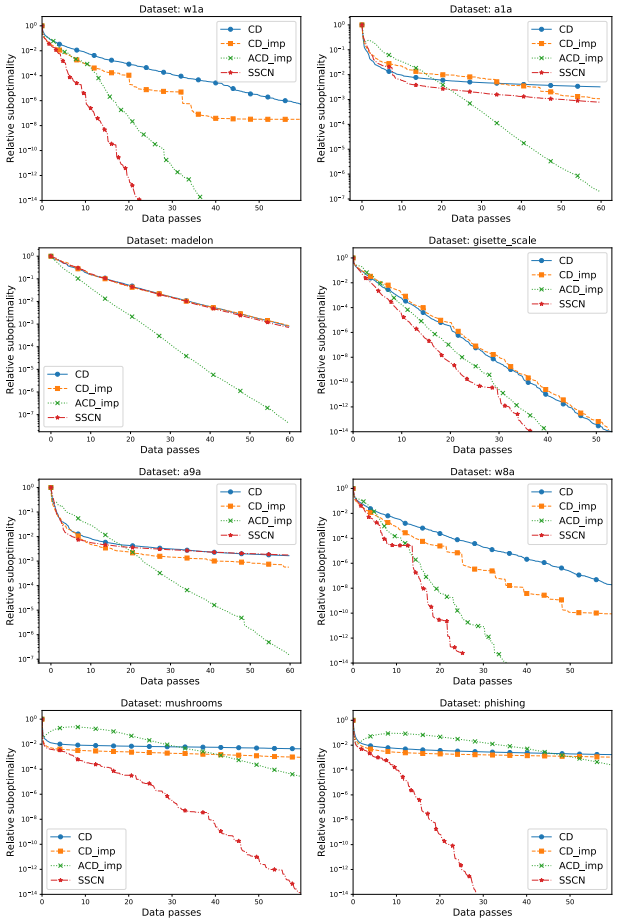

Figure 1. Comparison of CD with uniform sampling, CD with importance sampling, accelerated CD with importance sampling and SSCN with uniform sampling on LibSVM datasets.

In all examples, SSCN outperformed CD with uniform sampling. Moreover, the performance of SSCN was always either about the same or significantly better to CD with importance sampling. Furthermore, SSCN was also competitive to accelerated CD with importance sampling (in about half of the cases, SSCN was faster, while in the other half, accelerated CD was faster).

The next experiment studies the effect of $\tau(\mathbf{S})$ on the conver-

gence. We SSCN against fastest non-accelerated first-order method – SDNA, both with varying $\tau(\mathbf{S})$. Figure 2 presents the result. As expected, SSCN has outperformed SDNA.

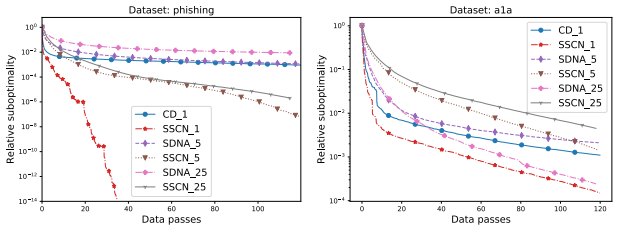

Figure 2. SSCN vs. SDNA on LibSVM datasets. All algorithms with uniform sampling. Legend indicates the values of $\tau(\mathbf{S})$.

9. Future Work

Lastly, we list several possible extensions of our work.

Acceleration. We believe it would be valuable to incorporate Nesterov's momentum into Algorithm 1. Ideally, one would like to get the global rate in between convergence rate of accelerated cubic regularized Newton (Nesterov, 2008) and accelerated CD (Allen-Zhu et al., 2016; Nesterov & Stich, 2017). On the other hand, the local rate (for strongly convex objectives) should recover accelerated sketch-and-project (Tu et al., 2017; Gower et al., 2018). If accelerated sketch-and-project is optimal (this is yet to be established), then accelerated SSCN (again, given that it recovers accelerated sketch-and-project) would be a locally optimal algorithm as well.

Non-separable ψ. As mentioned in Section 5.1, one should not hope for linear convergence of SSCN if ψ is not separable, as the iterates can "jump" away from the optimum in such case. This issue has been resolved for first-order methods using control variates (Hanzely et al., 2018), resulting in SEGA. Therefore, the development of second-order SEGA remains an interesting open problem.

Inexact method. SSCN is applicable in the setup, where function f is accessible via zeroth-order oracle only. In such a case, for any $\mathbf{S} \in \mathbb{R}^{\tau\times d}$ we can estimate $\nabla_{\mathbf{S}} f(x)$ and $\nabla^2_{\mathbf{S}} f(x)$ using $\mathcal{O}(\tau^2)$ function value evaluations. However, since both $\nabla_{\mathbf{S}} f(x)$ and $\nabla^2_{\mathbf{S}} f(x)$ are only evaluated inexactly, a slight modification of our theory is required.

Non-uniform sampling. Note that our local theory allows for arbitrary non-uniform distribution of $\mathbf{S}$, which might be potentially exploited. At the same time, in some applications, it might be feasible to use a greedy selection rule for $\mathbf{S}$ (our theory does not support that).

While developing optimal and implementable importance sampling for the local convergence is beyond the scope of this paper,[9] we sketch several possible sampling strategies that might yield faster convergence.[10]

- Let $\mathbb{P}(\mathbf{S} \in \{e_1, e_2, \dots, e_d\}) = 1$. If we evaluate the diagonal of the Hessian close to optimum (cost $\mathcal{O}(nd)$ for linear models) and sample proportionally to it, we obtain local linear rate with leading complexity term $\frac{\mathrm{Tr}\left(\nabla^2 f(x^*)\right)}{\lambda_{\min}\nabla^2 f(x^*)}$.
- It is unclear how to design an efficient importance sampling for minibatch (i.e., $1 < \mathbb{E}\left[\tau(\mathbf{S})\right] < d$) methods. Determinantal point processes (DPP) (Rodomanov & Kropotov, 2019; Mutný et al., 2019) were proposed to speed up SDNA from (Qu et al., 2016) (i.e., analogous CD with static matrix upper bound) – we thus believe they might be applicable on our setting too. However, in such a case, one would need to evaluate the whole Hessian close to optimum, which is infeasible for applications where d is large.
- It is known that SDNA (see related literature) is faster than minibatch CD under the ESO assumption (Qu & Richtárik, 2016a;b). Therefore, we might instead apply minibatch importance sampling for ESO assumption from (Hanzely & Richtárik, 2019) (which corresponds to optimizing the upper bound on iteration complexity). Using the mentioned sampling, we only require evaluating the diagonal of Hessian at some point close to optimum, which is of the same cost as computing the full gradient for linear models – thus is feasible.
- It is a natural question to ask whether one can speed up the convergence using a greedy rule instead of the random one. For standard CD, greedy rule was shown to have a superior iteration complexity to any randomized rule (Nutini et al., 2015; Karimireddy et al., 2019). For simplicity, consider case where $\mathbb{P}(\mathbf{S} \in \{e_1, e_2, \dots, e_d\}) = 1$. Far from the optimum, (approximate) greedy rule at iteration k chooses index $i = \mathrm{argmax}_j |\nabla_j f(x^k)|^{\frac{3}{2}} M_{e_j}^{-\frac{1}{2}}$. Close to optimum, if a diagonal of a Hessian was evaluated, (approximate) greedy index would be $\mathrm{argmax}_j |\nabla_j f(x^k)|^2 \nabla_{j,j} f(x)^{-1}$. For linear models, both of the mentioned cases are implementable using the efficient nearest neighbour search (Dhillon et al., 2011) with sublinear complexity in terms of d.

Acknowledgements

The work of the second and the fourth author was supported by ERC Advanced Grant 788368.

[9] As this is still an open problem even for sketch-and-project (Gower & Richtárik, 2015).

[10] This only applies to the local results as the global convergence requires some uniformity; see Assumption 5.1.

References

Allen-Zhu, Z., Qu, Z., Richtárik, P., and Yuan, Y. Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pp. 1110–1119, 2016.

Byrd, R. H., Chin, G. M., Neveitt, W., and Nocedal, J. On the use of stochastic Hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 21(3):977–995, 2011.

Carmon, Y. and Duchi, J. Gradient descent finds the cubic-regularized nonconvex Newton step. *SIAM Journal on Optimization*, 29(3):2146–2178, 2019.

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.

Chang, C.-C. and Lin, C.-J. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

Chang, K.-W., Hsieh, C.-J., and Lin, C.-J. Coordinate descent method for large-scale l2-loss linear support vector machines. *Journal of Machine Learning Research*, 9(Jul): 1369–1398, 2008.

Conn, A. R., Gould, N. I., and Toint, P. L. *Trust region methods*, volume 1. Siam, 2000.

Defazio, A., Bach, F., and Lacoste-Julien, S. 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.

Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Nearest neighbor based greedy coordinate descent. In *Advances in Neural Information Processing Systems*, pp. 2160–2168, 2011.

Doikov, N. and Nesterov, Y. Minimizing uniformly convex functions by cubic regularization of Newton method. *arXiv preprint arXiv:1905.02671*, 2019.

Doikov, N. and Richtárik, P. Randomized block cubic Newton method. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1290–1298, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/doikov18a.html.

Erdogdu, M. A. and Montanari, A. Convergence rates of sub-sampled Newton methods. In *Advances in Neural Information Processing Systems 28*, 2015.

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

Gower, R., Hanzely, F., Richtárik, P., and Stich, S. U. Accelerated stochastic matrix inversion: general theory and speeding up BFGS rules for faster second-order optimization. In *Advances in Neural Information Processing Systems*, pp. 1619–1629, 2018.

Gower, R. M. and Richtárik, P. Stochastic dual ascent for solving linear systems. *arXiv:1512.06890*, 2015.

Gower, R. M. and Richtárik, P. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1660–1690, 2015.

Gower, R. M. and Richtárik, P. Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1380–1409, 2017.

Gower, R. M., Kovalev, D., Lieder, F., and Richtárik, P. RSN: Randomized Subspace Newton. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 616–625. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8351-rsn-randomized-subspace-newton.pdf.

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

Grapiglia, G. N. and Nesterov, Y. Accelerated regularized Newton methods for minimizing composite convex functions. *SIAM Journal on Optimization*, 29(1):77–99, 2019.

Griewank, A. The modification of Newtons method for unconstrained optimization by bounding cubic terms. Technical report, Department of Applied Mathematics

and Theoretical Physics, University of Cambridge, 1981. Technical Report NA/12.

Hanzely, F. and Richtárik, P. Accelerated coordinate descent with arbitrary sampling and best rates for minibatches. In *Proceedings of Machine Learning Research*, pp. 304–312. PMLR, 2019.

Hanzely, F., Mishchenko, K., and Richtárik, P. SEGA: Variance reduction via gradient sketching. In *Advances in Neural Information Processing Systems*, pp. 2082–2093, 2018.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the Polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811. Springer, 2016.

Karimireddy, S. P., Stich, S. U., and Jaggi, M. Global linear convergence of Newton's method without strong-convexity or lipschitz gradients. *arXiv preprint arXiv:1806.00413*, 2018.

Karimireddy, S. P., Koloskova, A., Stich, S. U., and Jaggi, M. Efficient greedy coordinate descent for composite problems. In *International Conference on Artificial Intelligence and Statistics*, pp. 2887–2896, 2019.

Kohler, J. M. and Lucchi, A. Sub-sampled cubic regularization for non-convex optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1895–1904. JMLR. org, 2017.

Kovalev, D., Richtárik, P., Gorbunov, E., and Gasanov, E. Stochastic spectral and conjugate descent methods. In *Advances in Neural Information Processing Systems*, pp. 3358–3367, 2018.

Kovalev, D., Mishchenko, K., and Richtárik, P. Stochastic Newton and cubic Newton methods with simple local linear-quadratic rates. *arXiv preprint arXiv:1912.01597*, 2019.

Kozak, D., Becker, S., Doostan, A., and Tenorio, L. Stochastic subspace descent. *arXiv preprint arXiv:1904.01145*, 2019.

Monteiro, R. D. and Svaiter, B. F. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

Mutný, M., Dereziński, M., and Krause, A. Convergence analysis of the randomized Newton method with determinantal sampling. *arXiv preprint arXiv:1910.11561*, 2019.

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

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

Nesterov, Y. *Lectures on convex optimization*, volume 137. Springer, 2018.

Nesterov, Y. Inexact basic tensor methods. *CORE Discussion Papers 2019/23*, 2019.

Nesterov, Y. and Polyak, B. T. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Nesterov, Y. and Stich, S. U. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017.

Nutini, J., Schmidt, M., Laradji, I., Friedlander, M., and Koepke, H. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, pp. 1632–1641, 2015.

Pilanci, M. and Wainwright, M. J. Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization*, 27(1):205–245, 2017.

Qu, Z. and Richtárik, P. Coordinate descent with arbitrary sampling I: Algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016a.

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

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. Stochastic reformulations of linear systems: algorithms and convergence theory. *arXiv preprint arXiv:1706.01108*, 2017.

Rodomanov, A. and Kropotov, D. A randomized coordinate descent method with volume sampling. *arXiv preprint arXiv:1904.04587*, 2019.

Roosta-Khorasani, F. and Mahoney, M. W. Sub-sampled Newton methods. *Mathematical Programming*, 174(1-2): 293–326, 2019.

Roux, N. L., Schmidt, M., and Bach, F. R. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in neural information processing systems*, pp. 2663–2671, 2012.

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.

Stich, S. U., Muller, C. L., and Gartner, B. Optimization of convex functions with random pursuit. *SIAM Journal on Optimization*, 23(2):1284–1309, 2013.

Tripuraneni, N., Stern, M., Jin, C., Regier, J., and Jordan, M. I. Stochastic cubic regularization for fast nonconvex optimization. In *Advances in Neural Information Processing Systems*, pp. 2899–2908, 2018.

Tu, S., Venkataraman, S., Wilson, A. C., Gittens, A., Jordan, M. I., and Recht, B. Breaking locality accelerates block gauss-seidel. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3482–3491. JMLR. org, 2017.

Wang, Z., Zhou, Y., Liang, Y., and Lan, G. Stochastic variance-reduced cubic regularization for nonconvex optimization. *arXiv preprint arXiv:1802.07372*, 2018.

Wright, S. J. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

Xu, P., Yang, J., Roosta-Khorasani, F., Ré, C., and Mahoney, M. W. Sub-sampled Newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems*, pp. 3000–3008, 2016.

Xu, P., Roosta, F., and Mahoney, M. W. Newton-type methods for non-convex optimization under inexact hessian information. *Mathematical Programming*, pp. 1–36, 2017.

Appendix

A. Table of Frequently Used Notation

Table 1. Summary of frequently used notation.

From main paper		
$F:\mathbb{R}^d\to\mathbb{R}$	Objective function	(1)
$f:\mathbb{R}^d\to\mathbb{R}$	Smooth part of the objective	(1)
$\psi:\mathbb{R}^d\to\mathbb{R}\cup\{+\infty\}$	Non-smooth part of the objective	(1)
x^*	Global optimum of (1)	
F^*	$:=F(x^*)$, the optimum value of the objective	
$\mathbf{S}\in\mathbb{R}^{d,\tau(\mathbf{S})}$	Random matrix sampled from distribution $\mathcal{D}$	(2)
S	Random subset of $\{1,\dots,d\}$	(2)
μ	The constant of strong convexity	As. 5.9
$M_{\mathbf{S}}$	Lipschitz constant of $\nabla^2 f(x)$ on the range of $\mathbf{S}$	(4)
M	Lipschitz constant of $\nabla^2 f(x)$ on $\mathbb{R}^d$; $M=M_{\mathbf{I}^d}$	
L	Lipschitz constant of $\nabla f(x)$ on $\mathbb{R}^d$	
$\mathbf{A}_{\mathbf{S}}$	$:=\mathbf{S}^\top\mathbf{A}\mathbf{S}\in\mathbb{R}^{\tau(\mathbf{S})\times\tau(\mathbf{S})}$, for a given matrix $\mathbf{A}\in\mathbb{R}^{d\times d}$	
$\nabla_{\mathbf{S}}f(x)$	$:=\mathbf{S}^\top\nabla f(x)$	
$\nabla^2_{\mathbf{S}}f(x)$	$:=(\nabla^2 f(x))_{\mathbf{S}}=\mathbf{S}^\top\nabla^2 f(x)\mathbf{S}$	
$\mathbf{H}_{\mathbf{S}}(x)$	$:=\nabla^2_{\mathbf{S}}f(x)+\sqrt{\frac{M_{\mathbf{S}}}{2}}\|\nabla_{\mathbf{S}}f(x)\|^{\frac{1}{2}}\mathbf{I}^{\tau(\mathbf{S})}$	Lem. 6.1
ζ	$:=\lambda_{\min}\left(\left(\nabla^2 f(x^*)\right)^{\frac{1}{2}}\mathbb{E}\left[\mathbf{S}(\nabla^2_{\mathbf{S}}f(x^*))^{-1}\mathbf{S}^\top\right]\left(\nabla^2 f(x^*)\right)^{\frac{1}{2}}\right)$	(13)
$\mathbf{P}^{\mathbf{S}}$	$:=\mathbf{S}\left(\mathbf{S}^\top\mathbf{S}\right)^{-1}\mathbf{S}^\top$, the projection onto range of $\mathbf{S}$	Sec. 5.1
R	$:=\sup_{x\in\mathbb{R}^d}\left\{\|x-x^*\| \;:\; F(x)\le F(x^0)\right\}$	(9)
Standard		
$\mathbb{E}[\cdot]$	Expectation	
$\mathbb{P}(\cdot)$	Probability	
$\mathbf{I}^q$	Identity matrix in $\mathbb{R}^{q\times q}$	
$\lambda_{\max}(\cdot),\lambda_{\min}(\cdot)$	Maximal eigenvalue, minimal eigenvalue	
$\langle\cdot,\cdot\rangle$	Scalar product of vectors: $\langle x,y\rangle:=x^\top y$	
$\|\cdot\|$	Standard Euclidean norm: $\|x\|:=\sqrt{\langle x,x\rangle}$	
$\|\cdot\|_{\mathbf{B}}$	Weighted Euclidean norm: $\|x\|_{\mathbf{B}}:=\sqrt{\langle\mathbf{B}x,x\rangle}$	
e_i	i-th vector from the standard basis in $\mathbb{R}^d$	
e	Vector of ones in $\mathbb{R}^d$; i.e., $e:=\sum_{i=1}^d e_i$	
From Appendix		
$\lambda_f(x)$	$:=\left(\nabla f(x)^\top\left(\nabla^2 f(x)\right)^{-1}\nabla f(x)\right)^{\frac{1}{2}}$, Newton decrement	(22)
χ^0	$:=\{x;f(x)\le f(x^0)\}$, sublevel set	
$\mathrm{Tr}(\cdot)$	Trace	Sec. D.1

B. Extra Experiments

B.1. Logistic regression

Regularized logistic regression is a machine learning model for binary classification. Given data matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, labels $b \in \{-1,1\}^n$ and regularization parameter $\lambda \in \mathbb{R}_+$, the training corresponds to solving the following optimization problem

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} \log\left(1+\exp\left(\mathbf{A}_{i,:}x \cdot b\right)\right) + \frac{\lambda}{2}\|x\|^2.$$

In the first experiment, we compare SSCN to first-order coordinate descent (CD) on LIBSVM (Chang & Lin, 2011). We consider three different instances of CD: CD with uniform sampling, CD with importance sampling (Nesterov, 2012), and accelerated CD with importance sampling (Allen-Zhu et al., 2016; Nesterov & Stich, 2017).

In order to be comparable with the mentioned first-order methods, we consider $\mathbf{S} \in \{e_1,\dots,e_d\}$ with probability 1 – the complexity of performing each iteration is about the same for each algorithm now. At the same time, computing M_{e_i} for all $1 \le i \le d$ is of cost $\mathcal{O}(nd)$ – the same cost as computing coordinate-wise smoothness constants for (accelerated) CD (see Section 7.1 for the details). Figure 3 shows the result for non-normalized data, while Figure 4 shows the results for normalized data (thus importance sampling is identical to uniform).

In all examples, SSCN outperformed CD with uniform sampling. Moreover, the performance of SSCN was always either about the same or significantly better to CD with importance sampling. Furthermore, SSCN was also competitive to accelerated CD with importance sampling (in about half of the cases, SSCN was better, while in the other half, ACD was better).

In the second experiment, we compare methods with $\tau > 1$: SSCN and SDNA (Qu et al., 2016) (analogous first-order method). Again, we consider the logistic regression problem on LIBSVM data. We consider $\tau \in \{1,5,25\}$. In all cases, we sample uniformly – every subset of size τ have equal chance to be chosen at every iteration (independent of the past).

There is, however, one tricky part in terms of implementation. While we can evaluate and store M_{e_i} ($i \le d$) cheaply for linear models, this is not the case for evaluating/storing M_S (at least we do not know how to do it efficiently). Therefore, we use $M_S = M$ for $|S| > 1$ for SSCN. Figure 5 shows the result.

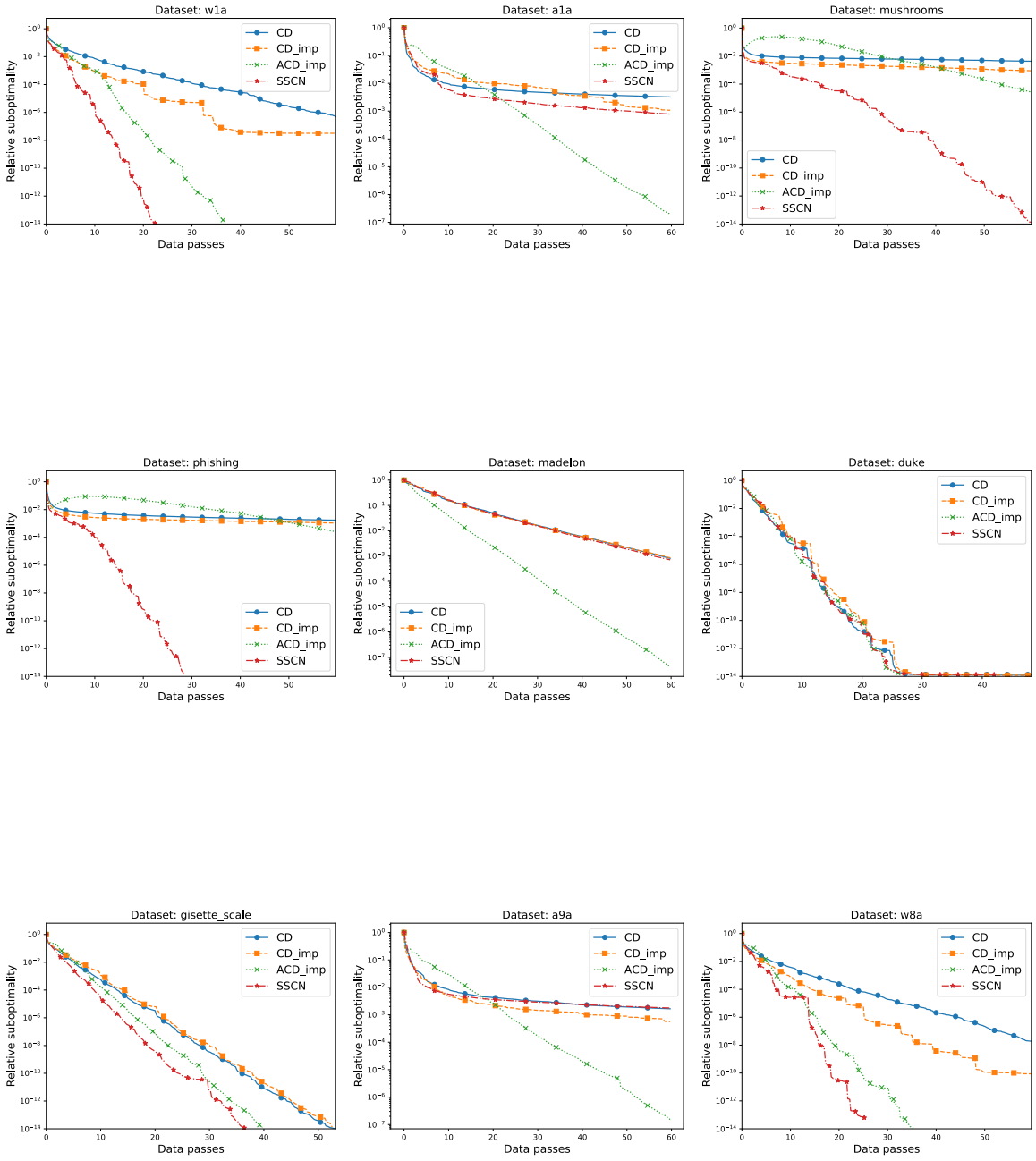

Figure 3. Comparison of CD with uniform sampling, CD with importance sampling, accelerated CD with importance sampling and SSCN (Algorithm 1) with uniform sampling on LibSVM datasets.

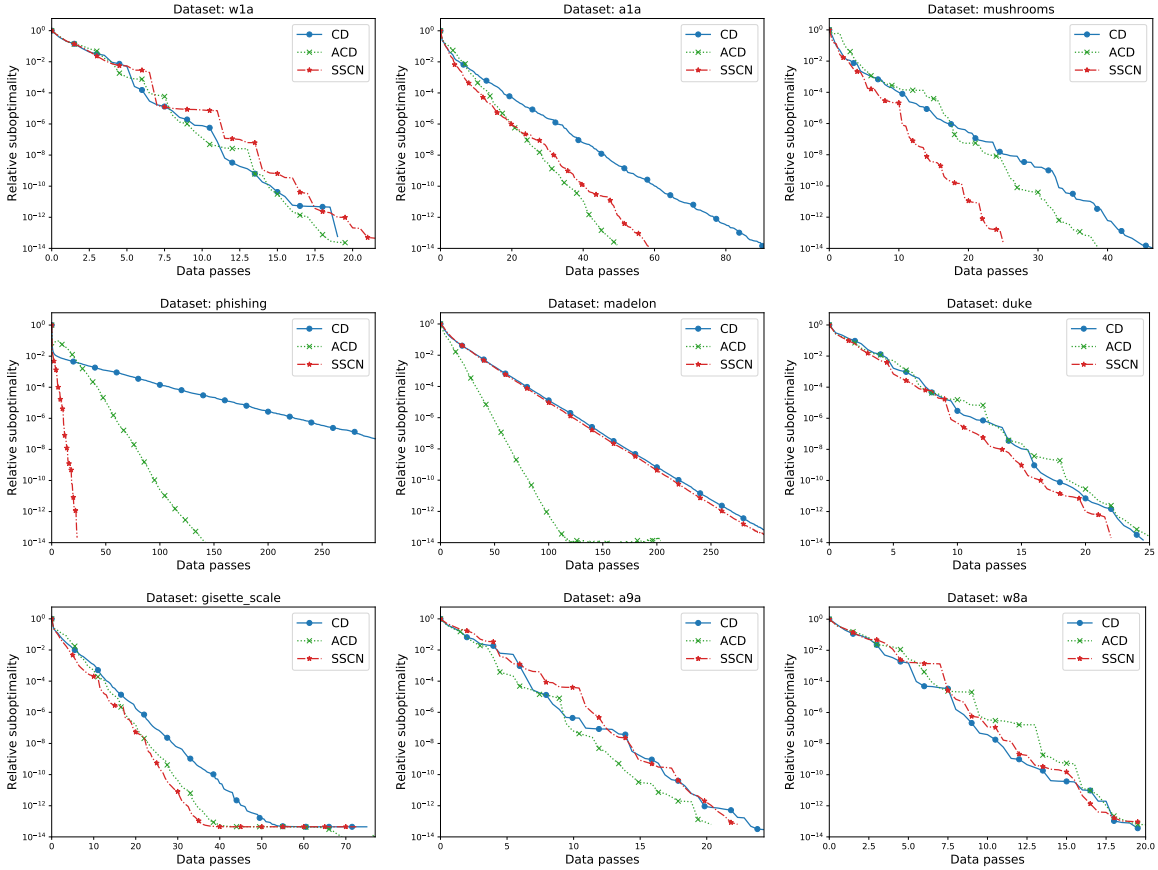

Figure 4. Comparison of coordinate descent, accelerated coordinate descent and SSCN (all with uniform sampling) on LibSVM datasets. In each case we have normalized the data matrix to have identical norms of all columns.

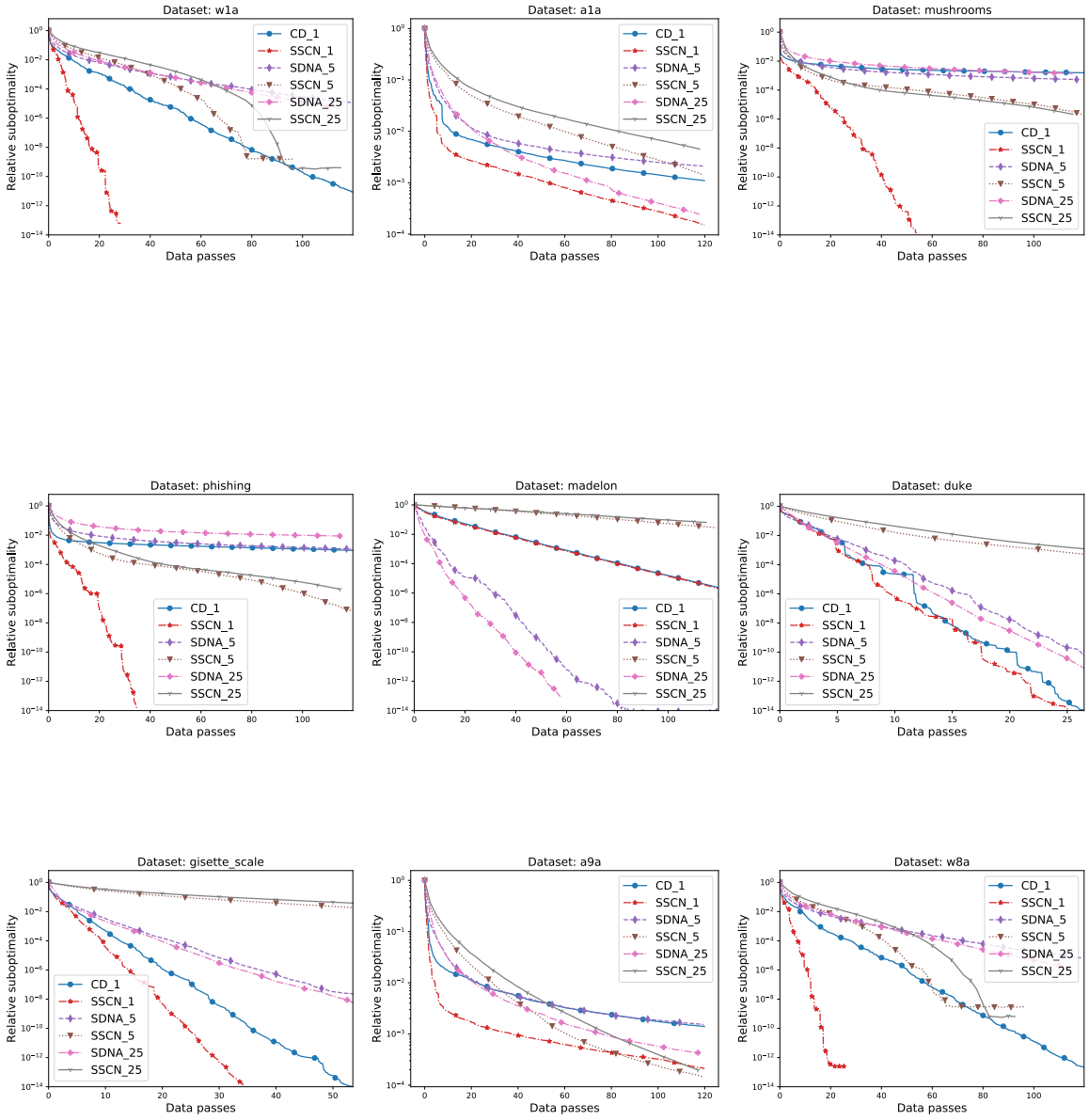

Figure 5. SSCN vs. SDNA on LibSVM datasets. All algorithms with uniform sampling.

B.2. Log-Sum-Exp

In this section, let us consider unconstrained minimization of the following Log-Sum-Exp function

$$f(x) \quad = \quad \sigma \log \left(\sum_{i=1}^{m} \exp\left(\tfrac{\langle a_i, x \rangle - b_i}{\sigma} \right) \right), \quad x \in \mathbb{R}^n,$$

where $\sigma > 0$ is a *smoothing* parameter, while $a_i \in \mathbb{R}^n, 1 \leq i \leq m$ and $b \in \mathbb{R}^m$ are given data. This function has both Lipschitz continuous gradient and Lipschitz continuous Hessian (see Example 1 in (Doikov & Nesterov, 2019)).

In our experiments, we first generate randomly elements of $\{\tilde{a}_i\}_{i=1}^m$ and b from uniform distribution on $[-1, 1]$. Then, we form an auxiliary function $\tilde{f}(x) := \sigma \log\Big(\sum_{i=1}^{m} \exp\big(\frac{\langle \tilde{a}_i, x \rangle - b_i}{\sigma} \big) \Big)$, using these parameters, and set

$$a_i \quad := \quad \tilde{a}_i - \nabla \tilde{f}(0), \quad 1 \leq i \leq m.$$

Thus, we essentially obtain the optimum x^* of f in the origin, since $\nabla f(0) = 0$. We use $x_0 := e$ (vector of all ones) as a starting point, and always set $m := 6n$.

For this problem, we compare the performance of SSCN with the first-order Coordinate Descent (CD), using uniform samples of coordinates $S \subseteq [n]$ of a fixed size $\tau = |S|$.

Note, that keeping scalar products $\{\langle a_i, x_k \rangle\}_{i=1}^m$ precomputed for a current point x_k, we are able to compute the partial gradient $\nabla_{\mathbf{S}} f(x^k)$ in time $O(\tau m)$ and the partial Hessian $\nabla^2_{\mathbf{S}} f(x^k)$ in time $O(\tau^2 m)$. To find the next direction h^k of SSCN (solving the Cubic subproblem), we call Nonlinear Conjugate Gradient method, and use the following condition as a stopping criterion:

$$\|\nabla_h T_{\mathbf{S}}(x^k; h^k)\| \quad \leq \quad 10^{-4},$$

where $T_{\mathbf{S}}(x^k; h) := \langle \nabla_{\mathbf{S}} f(x^k), h \rangle + \frac{1}{2} \langle \nabla^2_{\mathbf{S}} f(x^k) h, h \rangle + \frac{M_k}{6} \|\mathbf{S} h\|^3$ is the Cubic model, and $M_k \geq 0$ is a regularization constant.

For both methods, we use one-dimensional search at every iteration, to fit the corresponding parameter:

1. For the Coordinate Descent, we find L_k such that $f(x^k) - f(x^{k+1}) \geq \frac{1}{2L_k} \|\nabla_{\mathbf{S}} f(x^k)\|^2$, where x^{k+1} is the next point of the method: $x^{k+1} = x^k + \frac{1}{L_k} \mathbf{S} \nabla_{\mathbf{S}} f(x^k)$.
2. For SSCN, we find M_k such that (6) is satisfied, i.e. $f(x^k) - f(x^{k+1}) \geq -T_{\mathbf{S}}(x^k, h^k)$.

Therefore, we need to evaluate the function value inside the procedure, which is not very expensive.

The results are shown on Figures 6,7, for $n = 500$ and 1000 respectively[11]. We see, that SSCN outperforms CD significantly in terms of the iteration rate. For SSCN with a medium batchsize τ, we may obtain the best performance in terms of the total computational time.

[11] Clock time was evaluated using the machine with Intel Core i7-8700 CPU, 3.20GHz; 16 GB RAM.

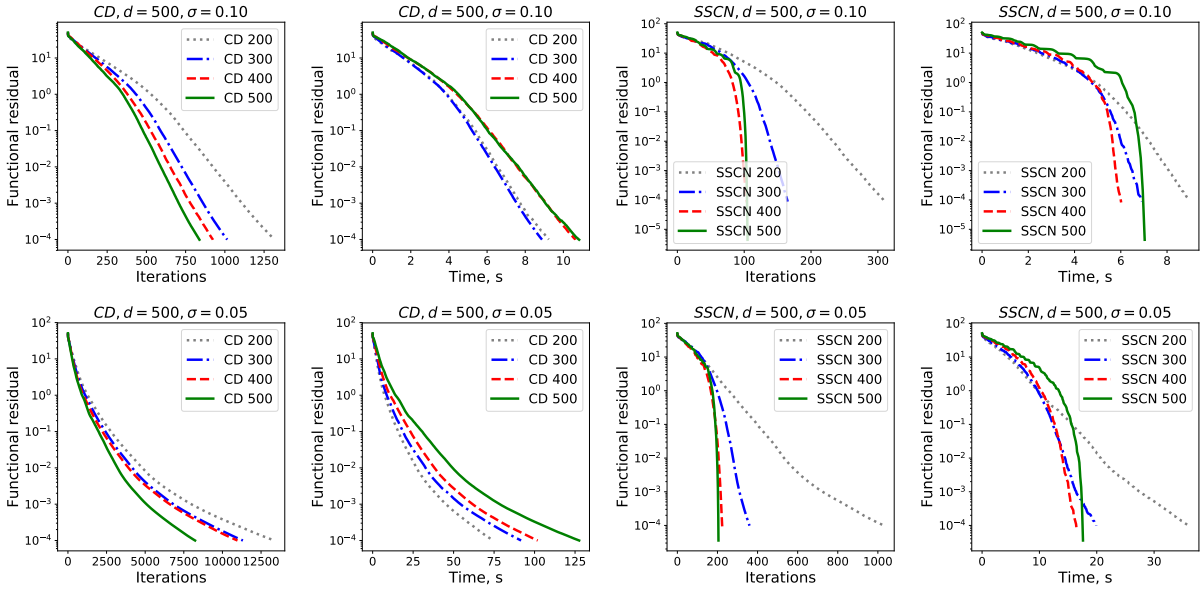

Figure 6. SSCN and Coordinate Descent (CD) methods, minimizing Log-Sum-Exp function, $n = 500$.

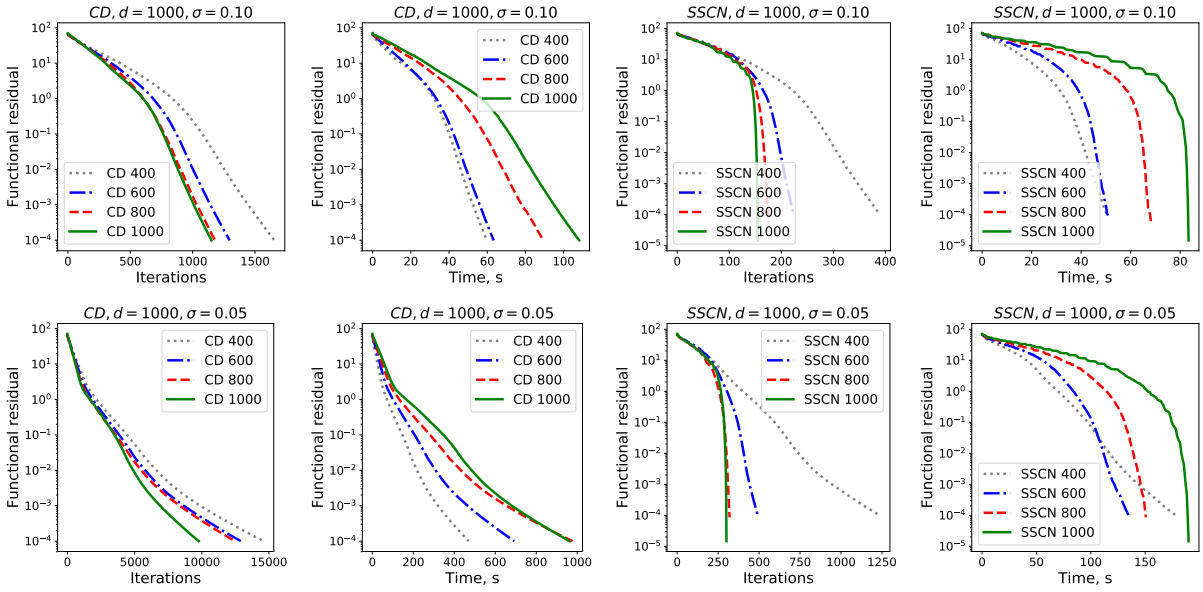

Figure 7. SSCN and Coordinate Descent (CD) methods, minimizing Log-Sum-Exp function, $n = 1000$.

C. Missing Proofs and Lemmas From Section 2

C.1. Explicit update

Lemma C.1. *Let* $x^+ = \operatorname{argmin}_y \langle g', y - x\rangle + \frac{H'}{2}\|x-y\|^2 + \frac{M'}{6}\|x-y\|^3$, *where* $H', M' > 0$. *Then we have*

$$x^+ = x - \frac{2g'}{H' + \sqrt{H'^2 + 2M'\|g'\|}} \tag{19}$$

Proof. By first-order optimality conditions we have $g' + H'(x^+ - x) + \frac{M'}{2}\|x^+ - x\|(x^+ - x) = 0$ which immediately yields

$$x^+ = x - \frac{g'}{H' + \frac{M'}{2}\|x^+ - x\|}. \tag{20}$$

Rearranging the terms and taking the norm we have $\frac{M'}{2}\|x^+ - x\|^2 + H'\|x^+ - x\| + \|g'\| = 0$. Solving the quadratic equation we arrive at

$$\|x^+ - x\| = \frac{\sqrt{H'^2 + 2M'\|g'\|} - H'}{M'}.$$

Plugging it back to (20), we get (19). □

C.2. Proof of Lemma 2.3

$$\begin{aligned}
&D_f(x^+, x) - \frac{1}{2}(x^+ - x)^\top \nabla^2 f(x)(x^+ - x) \\
&\quad = \int_0^1 \langle \nabla f(x + t(x^+ - x)) - f(x), x^+ - x\rangle\, dt - \frac{1}{2}(x^+ - x)^\top \nabla^2 f(x)(x^+ - x) \\
&\quad = \int_0^1 \int_0^1 \langle t\nabla^2 f(x + st(x^+ - x)), x^+ - x, x^+ - x\rangle\, ds\, dt - \frac{1}{2}(x^+ - x)^\top \nabla^2 f(x)(x^+ - x) \\
&\quad = \int_0^1 \int_0^1 \langle t\nabla^2 f(x + st(x^+ - x)) - t\nabla^2 f(x), x^+ - x, x^+ - x\rangle\, ds\, dt \\
&\quad = \int_0^1 \int_0^1 \int_0^1 \langle t^2 s\nabla^3 f(x + rst(x^+ - x)), x^+ - x, x^+ - x, x^+ - x\rangle\, dr\, ds\, dt.
\end{aligned}$$

Using (2) we get

$$\begin{aligned}
|f(x^+) - f(x) + \langle \nabla f(x), \mathbf{S}h\rangle + \frac{1}{2}h^\top \nabla^2_{\mathbf{S}} f(x) h| &\overset{(2)}{=} \left|\int_0^1 \int_0^1 \int_0^1 \langle t^2 s\nabla^3 f(x + rst\mathbf{S}h), \mathbf{S}h, \mathbf{S}h, \mathbf{S}h\rangle\, dr\, ds\, dt\right| \\
&\overset{(4)}{\leq} \int_0^1 \int_0^1 \int_0^1 t^2 s M_{\mathbf{S}} \|h_{\mathbf{S}}\|^3\, dr\, ds\, dt \\
&= \frac{M_{\mathbf{S}}}{6}\|h_{\mathbf{S}}\|^3.
\end{aligned}$$

C.3. Proof of Lemma 2.2

First, $M \geq M_{\mathbf{S}}$ is trivial. At the same time $M = M_{\mathbf{S}}$ if $\nabla^3 f(x)$ is identity tensor always, which corresponds to $f(x) = \frac{1}{6}\sum_{i=1}^d x_i^3$. Therefore, the inequality is tight.

To show sharpness of $M_{\mathbf{S}} \geq \left(\frac{\tau}{d}\right)^{\frac{3}{2}} M$, consider $f(x) = \frac{1}{6}(x^\top e)^3$. In this case, we have[12] $\nabla^3 f(x) = [e]^3$ and $\mathbf{S} = e_i$. In such case, $M = d^{\frac{3}{2}}$ and $M_{\mathbf{S}} = \tau^{\frac{3}{2}}$.

Note that f is non-convex in both examples. However, it is is convex on a set where $x_i \geq 0$ for all i.

[12] By $[e] \in \mathbb{R}^{d\times d\times d}$ we mean third order outer product of vector e.

D. Proofs for Section 5

D.1. Proof of Lemma 5.2

Let $\mathrm{Tr}\left(\mathbf{A}\right)$ be a trace of square matrix $\mathbf{A}$. We have

$$\begin{aligned}\mathbb{E}\left[\tau(\mathbf{S})\right] &= \mathbb{E}\left[\mathrm{Tr}\left(\mathbf{I}^{\tau(\mathbf{S})}\right)\right] = \mathbb{E}\left[\mathrm{Tr}\left(\mathbf{S}^\top\mathbf{S}\left(\mathbf{S}^\top\mathbf{S}\right)^{-1}\right)\right] = \mathbb{E}\left[\mathrm{Tr}\left(\mathbf{S}\left(\mathbf{S}^\top\mathbf{S}\right)^{-1}\mathbf{S}^\top\right)\right]\\ &= \mathrm{Tr}\left(\mathbb{E}\left[\mathbf{S}\left(\mathbf{S}^\top\mathbf{S}\right)^{-1}\mathbf{S}^\top\right]\right) \overset{(7)}{=} \mathrm{Tr}\left(\frac{\tau}{d}\mathbf{I}^d\right)\\ &= \tau.\end{aligned}$$

D.2. Proof of Lemma 5.7

For any $h' \in \mathbb{R}^d$ denote

$$\Omega_{\mathbf{S}}(x,h') \overset{\text{def}}{=} \langle\nabla f(x), \mathbf{P}^{\mathbf{S}}h'\rangle + \frac{1}{2}\langle\nabla^2 f(x)\mathbf{P}^{\mathbf{S}}h', \mathbf{P}^{\mathbf{S}}h'\rangle + \frac{M_{\mathbf{S}}}{6}\|\mathbf{P}^{\mathbf{S}}h'\|^3 + \psi(x+\mathbf{P}^{\mathbf{S}}h').$$

Clearly, it holds

$$\min_{h'\in\mathbb{R}^d}\Omega_{\mathbf{S}}(x,h') = \min_{h\in\mathbb{R}^{\tau(S)}}T_{\mathbf{S}}(x,h).$$

Therefore, for any fixed $y \in \mathbb{R}^d$ we have

$$F(x^{k+1}) \overset{(6)}{\leq} f(x^k) + \min_{h'\in\mathbb{R}^d}\Omega_{\mathbf{S}}(x^k,h') \leq f(x^k) + \Omega_{\mathbf{S}}(x^k; y-x^k).$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[F(x^{k+1})\,|\,x^k\right] &\leq f(x^k) + \mathbb{E}\left[\Omega_{\mathbf{S}}(x^k; y-x^k)\right]\\ &= f(x^k) + \frac{\tau}{d}\langle\nabla f(x^k), y-x^k\rangle + \mathbb{E}\left[\frac{1}{2}\langle\mathbf{P}^{\mathbf{S}}\nabla^2 f(x^k)\mathbf{P}^{\mathbf{S}}(y-x^k), y-x^k\rangle\right]\\ &\quad + \frac{M}{6}\mathbb{E}\left[\|\mathbf{P}^{\mathbf{S}}(y-x^k)\|^3\right] + \mathbb{E}\left[\psi(x^k+\mathbf{P}^{\mathbf{S}}(y-x^k))\right].\end{aligned}$$

Let us get rid of the expectations above. Firstly, we have

$$\begin{aligned}\mathbb{E}\left[\psi(x+\mathbf{P}^{\mathbf{S}}(y-x^k))\right] &= \mathbb{E}\left[\left\langle\psi'\left(\left(\mathbf{I}^d-\mathbf{P}^{\mathbf{S}}\right)x^k+\mathbf{P}^{\mathbf{S}}y\right), e\right\rangle\right]\\ &= \mathbb{E}\left[\left\langle\left(\mathbf{I}^d-\mathbf{P}^{\mathbf{S}}\right)\psi'\left(x^k\right), e\right\rangle\right] + \mathbb{E}\left[\left\langle\mathbf{P}^{\mathbf{S}}\psi'\left(y\right), e\right\rangle\right]\\ &= \left(1-\frac{\tau}{d}\right)\psi(x^k) + \frac{\tau}{d}\psi(y).\end{aligned}$$

For the cubed norm it can be estimated as follows

$$\mathbb{E}\left[\|\mathbf{P}^{\mathbf{S}}h'\|^3\right] \leq \|h'\|\cdot\mathbb{E}\left[\|\mathbf{P}^{\mathbf{S}}h'\|^2\right] = \frac{\tau}{d}\|h'\|^3, \qquad \forall h'\in\mathbb{R}^d.$$

Lastly, note that

$$\begin{aligned}\mathbb{E}\left[\mathbf{P}^{\mathbf{S}}\nabla^2 f(x^k)\mathbf{P}^{\mathbf{S}}\right] &= \mathbb{E}\left[\mathbf{P}^{\mathbf{S}}\left(\nabla^2 f(x^k)\right)^{\frac{1}{2}}\right]\mathbb{E}\left[\left(\nabla^2 f(x^k)\right)^{\frac{1}{2}}\mathbf{P}^{\mathbf{S}}\right]\\ &\quad + \mathbb{E}\left[\left(\mathbf{P}^{\mathbf{S}}\left(\nabla^2 f(x^k)\right)^{\frac{1}{2}} - \mathbb{E}\left[\mathbf{P}^{\mathbf{S}}\left(\nabla^2 f(x^k)\right)^{\frac{1}{2}}\right]\right)\left(\mathbf{P}^{\mathbf{S}}\left(\nabla^2 f(x^k)\right)^{\frac{1}{2}} - \mathbb{E}\left[\mathbf{P}^{\mathbf{S}}\left(\nabla^2 f(x^k)\right)^{\frac{1}{2}}\right]\right)^\top\right]\\ &= \frac{\tau^2}{d^2}\nabla^2 f(x^k) + \mathbb{E}\left[\left(\mathbf{P}^{\mathbf{S}}-\frac{\tau}{d}\mathbf{I}^d\right)\nabla^2 f(x^k)\left(\mathbf{P}^{\mathbf{S}}-\frac{\tau}{d}\mathbf{I}^d\right)\right]\\ &\preceq \frac{\tau^2}{d^2}\nabla^2 f(x^k) + L\mathbb{E}\left[\left(\mathbf{P}^{\mathbf{S}}-\frac{\tau}{d}\mathbf{I}^d\right)^2\right]\\ &= \frac{\tau^2}{d^2}\nabla^2 f(x^k) + \frac{\tau(d-\tau)}{d^2}L\mathbf{I}^d.\end{aligned}$$

Therefore, we conclude

$$\begin{aligned}\mathbb{E}\left[F(x^{k+1})\,|\,x^k\right] &\leq f(x^k)+\frac{\tau}{d}\langle\nabla f(x^k),y-x^k\rangle+\frac{\tau(d-\tau)}{d^2}\cdot\frac{L}{2}\|y-x^k\|^2\\ &+\frac{\tau^2}{d^2}\cdot\frac{1}{2}\langle\nabla^2 f(x^k)(y-x^k),y-x^k\rangle+\frac{\tau}{d}\cdot\frac{M}{6}\|y-x^k\|^3\\ &+\frac{\tau}{d}\psi(y)+\left(1-\frac{\tau}{d}\right)\psi(x^k).\end{aligned}$$

Finally, by convexity and from Lipschitz continuity of the Hessian (5), we have the following upper estimate:

$$\begin{aligned}&\langle\nabla f(x^k),y-x^k\rangle+\frac{\tau}{d}\cdot\frac{1}{2}\langle\nabla^2 f(x^k)(y-x^k),y-x^k\rangle\\ &\quad=\frac{d-\tau}{d}\langle\nabla f(x^k),y-x^k\rangle+\frac{\tau}{d}\Big(\langle\nabla f(x^k),y-x^k\rangle+\frac{1}{2}\langle\nabla^2 f(x^k)(y-x^k),y-x^k\rangle\Big)\\ &\quad\leq\frac{d-\tau}{d}\Big(f(y)-f(x^k)\Big)+\frac{\tau}{d}\Big(f(y)-f(x^k)+\frac{M}{6}\|y-x^k\|^3\Big)\\ &\quad\leq f(y)-f(x^k)+\frac{M}{6}\|y-x^k\|^3.\end{aligned}$$

which completes the proof. □

D.3. Proof of Theorem 5.8

Let us denote the following auxiliary sequences:

$$a_k \overset{\text{def}}{=} k^2, \qquad A_k \overset{\text{def}}{=} A_0+\sum_{i=1}^{k}a_i, \qquad k\geq 1,$$

and

$$A_0 \overset{\text{def}}{=} \frac{4}{3}\left(\frac{d}{\tau}\right)^3.$$

Then, we have an estimate

$$A_k = A_0+\sum_{i=1}^{k}i^2 \geq A_0+\int_0^k x^2dx = A_0+\frac{k^3}{3}. \tag{21}$$

Now, let us fix iteration counter $k\geq 0$ and set

$$\alpha_k \overset{\text{def}}{=} \frac{d}{\tau}\frac{a_{k+1}}{A_{k+1}} \quad\Leftrightarrow\quad 1-\frac{\tau}{d}\alpha_k = \frac{A_k}{A_{k+1}}.$$

Note that we have $\alpha_k\leq 1$ by the choice of A_0, since it holds

$$\max_{\xi\geq 0}\frac{\xi^2}{A_0+\frac{\xi^3}{3}} = \frac{\tau}{d}.$$

Let us plug $y \equiv \alpha_k x^* + (1-\alpha_k)x^k$ into (8). By convexity we obtain

$$\begin{aligned}
\mathbb{E}\left[F(x^{k+1}) \,|\, x^k\right] &\leq \left(1-\frac{\tau}{d}\right)F(x^k) + \frac{\tau}{d}\alpha_k F^* + \frac{\tau}{d}(1-\alpha_k)F(x^k) \\
&\quad + \frac{\tau}{d}\left(\frac{d-\tau}{d}\frac{L\|x^k-x^*\|^2}{2}\alpha_k^2 + \frac{M\|x^k-x^*\|^3}{3}\alpha_k^3\right) \\
&= \frac{A_k}{A_{k+1}}F(x^k) + \frac{a_{k+1}}{A_{k+1}}F^* + \frac{d}{\tau}\frac{d-\tau}{d}\frac{L\|x^k-x^*\|^2}{2}\left(\frac{a_{k+1}}{A_{k+1}}\right)^2 + \left(\frac{d}{\tau}\right)^2\frac{M\|x^k-x^*\|^3}{3}\left(\frac{a_{k+1}}{A_{k+1}}\right)^3 \\
&\leq \frac{A_k}{A_{k+1}}F(x^k) + \frac{a_{k+1}}{A_{k+1}}F^* + \frac{d-\tau}{\tau}\frac{LR^2}{2}\left(\frac{a_{k+1}}{A_{k+1}}\right)^2 + \left(\frac{d}{\tau}\right)^2\frac{MR^3}{3}\left(\frac{a_{k+1}}{A_{k+1}}\right)^3.
\end{aligned}$$

Therefore, for the residual $\delta_k \overset{\text{def}}{=} \mathbb{E}\left[F(x^k)\right] - F^*$ we have the following bound

$$A_{k+1}\delta_{k+1} \leq A_k\delta_k + \frac{d-\tau}{\tau}\frac{LR^2}{2}\frac{a_{k+1}^2}{A_{k+1}} + \left(\frac{d}{\tau}\right)^2\frac{MR^3}{3}\frac{a_{k+1}^3}{A_{k+1}^2}, \quad k \geq 0.$$

Summing up these inequalities for different k, we obtain

$$A_k\delta_k \leq A_0\delta_0 + \frac{d-\tau}{\tau}\frac{LR^2}{2}\sum_{i=1}^k\frac{a_i^2}{A_i} + \left(\frac{d}{\tau}\right)^2\frac{MR^3}{3}\sum_{i=1}^k\frac{a_i^3}{A_i^2}, \qquad k \geq 1.$$

To finish the proof it remains to notice that

$$\sum_{i=1}^k\frac{a_i^2}{A_i} \overset{(21)}{\leq} \sum_{i=1}^k\frac{i^4}{A_0+\frac{1}{3}i^3} \leq 3\sum_{i=1}^k i \leq 3k^2,$$

and

$$\sum_{i=1}^k\frac{a_i^3}{A_i^2} \overset{(21)}{\leq} \sum_{i=1}^k\frac{i^6}{(A_0+\frac{1}{3}i^3)^2} \leq 9k.$$

□

D.4. Proof of Theorem 5.10

Given that Assumption 5.9 (strong convexity) is satisfied, the following inequality holds

$$\frac{\mu}{2}\|x-x^*\|^2 \leq F(x)-F^*, \qquad \forall x \in \mathbb{R}^d,$$

and thus we have a bound for the radius of level sets (9):

$$R^2 \leq \frac{2}{\mu}(F(x^0)-F^*).$$

Combining the above with (10) we obtain the following convergence estimate:

$$\mathbb{E}\left[F(x^k)-F^*\right] \leq \left(\frac{d-\tau}{\tau}\cdot\frac{18L}{\mu k} + \left(\frac{d}{\tau}\right)^2\cdot\frac{18MR}{\mu k^2} + \frac{1}{1+\frac{1}{4}\left(\frac{\tau}{d}k\right)^3}\right)\cdot\left(F(x^0)-F^*\right), \quad k \geq 1.$$

Therefore, we get the linear decrease of the expected residual

$$\mathbb{E}\left[F(x^k)-F^*\right] \leq \frac{1}{2}\left(F(x^0)-F^*\right),$$

as soon as the following three bounds for k are all reached:

1. $\frac{d-\tau}{\tau}\cdot\frac{18L}{\mu k}\leq\frac{1}{6}\quad\Leftrightarrow\quad k\geq 108\frac{d-\tau}{\tau}\cdot\frac{L}{\mu}$.
2. $\left(\frac{d}{\tau}\right)^2\cdot\frac{18MR}{\mu k^2}\leq\frac{1}{6}\quad\Leftrightarrow\quad k\geq\frac{d}{\tau}\sqrt{108\frac{MR}{\mu}}$.
3. $\frac{1}{1+\frac{1}{4}\left(\frac{\tau}{d}k^3\right)^3}\leq\frac{1}{6}\quad\Leftrightarrow\quad k\geq\frac{d}{\tau}20^{1/3}$.

□

E. Proofs for Section 6

E.1. Several technical Lemmas

It will be convenient to denote the Newton decrement as follows:

$$\lambda_f(x):=\left(\nabla f(x)^\top\left(\nabla^2 f(x)\right)^{-1}\nabla f(x)\right)^{\frac{1}{2}} \tag{22}$$

and a sublevel set of x^0 as χ^0; i.e., $\chi^0:=\{x;f(x)\leq f(x^0)\}$.

Lemma E.1. *(Local bounds) Suppose that x^0 is such that $f(x^0)-f(x^*)\leq\varrho^4\frac{2\left(\min_{x\in\chi^0}\lambda_{\min}\nabla^2_{\mathbf{S}}f(x)\right)^4}{LM^2_{\mathbf{S}}\|\mathbf{S}\|^2}$ for some $\varrho>0$. Then, we have*

$$\sqrt{\frac{M_{\mathbf{S}}}{2}}\|\mathbf{S}^\top\nabla f(x^k)\|^{\frac{1}{2}}\mathbf{I}^{\tau(\mathbf{S})}\preceq\varrho\nabla^2_{\mathbf{S}}f(x^k). \tag{23}$$

Suppose further that $f(x^0)-f(x^*)\leq\varphi^2\frac{\mu\left(\lambda_{\min}\nabla^2_{\mathbf{S}}f(x^*)\right)^2}{2M^2_{\mathbf{S}}}$ for some $\varphi>0$. Then we have

$$(1-\varphi)\nabla^2_{\mathbf{S}}f(x^*)\preceq\nabla^2_{\mathbf{S}}f(x^k)\preceq(1+\varphi)\nabla^2_{\mathbf{S}}f(x^*). \tag{24}$$

Lastly, if $f(x^0)-f(x^*)\leq\omega^{-1}\left(\frac{2\mu^{\frac{3}{2}}}{(1+\gamma^{-1})M}\right)$ where $\omega(y):=y-\log(1+y)$ and $\gamma>0$, we have

$$f(x^k)-f(x^*)\leq\frac{1}{2}(1+\gamma)\lambda_f(x^k)^2. \tag{25}$$

Proof. For the sake of simplicity, let $x=x^k$ and $\mathbf{S}=\mathbf{S}^k$ throughout this proof. For the first part, we have

$$\begin{aligned}\sqrt{\frac{M_{\mathbf{S}}}{2}}\|\mathbf{S}^\top\nabla f(x)\|^{\frac{1}{2}}\mathbf{I}^{\tau(\mathbf{S})} &\preceq \sqrt{\frac{M_{\mathbf{S}}}{2}}\|\mathbf{S}\|^{\frac{1}{2}}\|\nabla f(x)\|^{\frac{1}{2}}\mathbf{I}^{\tau(\mathbf{S})}\\ &\preceq \sqrt{\frac{M_{\mathbf{S}}}{2}}\|\mathbf{S}\|^{\frac{1}{2}}2^{\frac{1}{4}}L^{\frac{1}{4}}\left(f(x^0)-f(x^*)\right)^{\frac{1}{4}}\mathbf{I}^{\tau(\mathbf{S})}\\ &\preceq \varrho\min_{x\in\chi^0}\lambda_{\min}\nabla^2_{\mathbf{S}}f(x)\mathbf{I}^{\tau(\mathbf{S})}\ \preceq\ \varrho\nabla^2_{\mathbf{S}}f(x).\end{aligned}$$

For the second part, we have

$$\begin{aligned}\nabla^2_{\mathbf{S}}f(x)-\nabla^2_{\mathbf{S}}f(x^*) &\preceq M_{\mathbf{S}}\|x-x^*\|\mathbf{I}^{\tau(\mathbf{S})}\\ &\preceq M_{\mathbf{S}}\sqrt{\frac{2(f(x)-f(x^*))}{\mu}}\mathbf{I}^{\tau(\mathbf{S})}\\ &\preceq M_{\mathbf{S}}\sqrt{\frac{2(f(x^0)-f(x^*))}{\mu}}\mathbf{I}^{\tau(\mathbf{S})}\\ &\preceq \varphi\nabla^2_{\mathbf{S}}f(x^*).\end{aligned}$$

Therefore, we can conclude that $\nabla^2_{\mathbf{S}}f(x)\preceq(1+\varphi)\nabla^2_{\mathbf{S}}f(x^*)$. Analogously we can show $\nabla^2_{\mathbf{S}}f(x^*)\preceq(1-\varphi)^{-1}\nabla^2_{\mathbf{S}}f(x)$ and thus (24) follows.

Lastly, if $f(x^0) - f(x^*) \leq \omega\left(\frac{2\mu^{\frac{3}{2}}}{(1+\gamma^{-1})M}\right)$, then due to (Nesterov, 2018) we have

$$\omega\left(\lambda_f(x)\right) \leq f(x) - f(x^*) \leq f(x^0) - f(x^*) \leq \omega\left(\frac{2\mu^{\frac{3}{2}}}{(1+\gamma^{-1})M}\right)$$

and thus $\lambda_f(x) \leq \frac{2\mu^{\frac{3}{2}}}{(1+\gamma^{-1})M}$. Now (25) follows from Lemma E.2 and Lemma E.3. □

Lemma E.2. *Function f is $\frac{M}{2\mu^{\frac{3}{2}}}$ self-concordant.*

Proof. See Example 5.1.1 in (Nesterov, 2018). □

Lemma E.3. *Consider any $\gamma \in \mathbb{R}^+$ and suppose that f is ς self-concordant. Then if $\lambda_f(x) < \frac{2}{(1+\gamma^{-1})\varsigma}$ we have*

$$f(x) - f(x^*) \leq \frac{1}{2}\left(1+\gamma\right)\lambda_f(x)^2 \tag{26}$$

Proof. Define $\omega_*(z) := -z - \ln(1-z)$. Note first that, $h(x) := \frac{\varsigma^2}{4}f(x)$ is 2 self concordant (Nesterov, 2018). As a consequence, if $\lambda_h(x) < 1$ we have (Nesterov, 2018)

$$h(x) - h(x^*) \leq \omega_*(\lambda_h(x)).$$

If further $\lambda_h(x) \leq \frac{1}{1+\gamma^{-1}}$ due to Lemma E.4, we get

$$\omega_*(\lambda_h(x)) \leq (1+\gamma)\frac{\lambda_h(x)^2}{2}.$$

As $\lambda_h(x) = \frac{\varsigma}{2}\lambda_f(x)$, we get (26). □

Lemma E.4. *Let $c \in \mathbb{R}^+$ and $0 \leq y \leq \frac{1}{1+c}$. Then we have $\omega_*(y) \leq \left(1+\frac{1}{c}\right)\frac{y^2}{2}$.*

Proof. See Lemma 5.1.5 in (Nesterov, 2018). □

E.2. Proof of Lemma 6.1

Note that the update rule of SSCN yields immediately (using first-order optimality conditions)

$$-\mathbf{S}^\top \nabla f(x) = \left(\nabla^2_{\mathbf{S}} f(x) + \frac{1}{2}M_{\mathbf{S}}\|x^+ - x\|\mathbf{I}^{\tau(\mathbf{S})}\right)\left(x^+ - x\right) \tag{27}$$

and therefore

$$\begin{aligned}
\left\|\mathbf{S}^\top \nabla f(x)\right\|^{\frac{1}{2}} &= \left(\left(x^+ - x\right)^\top \left(\nabla^2_{\mathbf{S}} f(x) + \frac{1}{2}M_{\mathbf{S}}\|x^+ - x\|\mathbf{I}^{\tau(\mathbf{S})}\right)^2 \left(x^+ - x\right)\right)^{\frac{1}{4}} \\
&\geq \left(\left(x^+ - x\right)^\top \left(\frac{1}{2}M_{\mathbf{S}}\|x^+ - x\|\mathbf{I}^{\tau(\mathbf{S})}\right)^2 \left(x^+ - x\right)\right)^{\frac{1}{4}} \\
&= \sqrt{\frac{M_{\mathbf{S}}}{2}}\|x^+ - x\|.
\end{aligned} \tag{28}$$

Furthermore, taking dot product of (27) with $(x^+ - x)$ yields

$$\left\langle \mathbf{S}^\top \nabla f(x), x^+ - x\right\rangle + \left\langle \nabla^2_{\mathbf{S}} f(x)\left(x^+ - x\right), x^+ - x\right\rangle + \frac{1}{2}M_{\mathbf{S}}\|x^+ - x\|^3 = 0$$

and thus

$$\begin{aligned}
f(x)-f(x^+) &\overset{(5)}{\geq} \left\langle \mathbf{S}^\top \nabla f(x), x-x^+\right\rangle - \frac{1}{2}\left\langle \nabla^2_{\mathbf{S}} f(x)(x^+-x), x^+-x\right\rangle - \frac{M_{\mathbf{S}}}{6}\|x^+-x\|^3 \\
&= \frac{1}{2}\left\langle \nabla^2_{\mathbf{S}} f(x)(x^+-x), x^+-x\right\rangle + \frac{M_{\mathbf{S}}}{3}\|x^+-x\|^3 \\
&\overset{(*)}{\geq} \frac{1}{2}\left(x^+-x\right)^\top \left(\nabla^2_{\mathbf{S}} f(x) + \frac{1}{2}M_{\mathbf{S}}\|x^+-x\|\mathbf{I}^{\tau(\mathbf{S})}\right)\left(x^+-x\right) \\
&\overset{(27)}{=} \frac{1}{2}\nabla f(x)^\top \mathbf{S}\left(\nabla^2_{\mathbf{S}} f(x) + \frac{1}{2}M_{\mathbf{S}}\|x^+-x\|\mathbf{I}^{\tau(\mathbf{S})}\right)^{-1}\mathbf{S}^\top \nabla f(x) \\
&\overset{(28)}{\geq} \frac{1}{2}\nabla f(x)^\top \mathbf{S}\left(\nabla^2_{\mathbf{S}} f(x) + \sqrt{\frac{M_{\mathbf{S}}}{2}}\|\mathbf{S}^\top \nabla f(x)\|^{\frac{1}{2}}\mathbf{I}^{\tau(\mathbf{S})}\right)^{-1}\mathbf{S}^\top \nabla f(x).
\end{aligned}$$

Above, in inequality $(*)$ we have used the fact that matrix $\left(\nabla^2_{\mathbf{S}} f(x) + \frac{1}{2}M_{\mathbf{S}}\|x^+-x\|\mathbf{I}^{\tau(\mathbf{S})}\right)$ is invertible since f is strongly convex and thus $\nabla^2_{\mathbf{S}} f(x) \succ 0$.

E.3. Proof of Theorem 6.2

First, suppose that $f(x^0)-f(x^*) \leq \varrho^4 \frac{2\left(\min_{x\in\mathcal{X}^0}\lambda_{\min}\nabla^2_{\mathbf{S}} f(x^k)\right)^4}{LM^2_{\mathbf{S}}\|\mathbf{S}\|^2}$ for some $\varrho>0$. Using the fact that $\nabla^2_{\mathbf{S}} f(x^k)$ is invertible ($\mathbf{S}$ has full column rank and $\nabla^2 f(x^k)\succ 0$) we have

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{2}\|\mathbf{S}^\top \nabla f(x^k)\|^2_{(\mathbf{H}(x^k))^{-1}}\right] &\overset{(23)}{\geq} \mathbb{E}\left[\frac{1}{2}\nabla f(x^k)^\top \mathbf{S}\left((1+\varrho)\nabla^2_{\mathbf{S}} f(x^k)\right)^{-1}\mathbf{S}^\top \nabla f(x^k)\right] \\
&= \frac{1}{2(1+\varrho)}\nabla f(x^k)^\top \mathbb{E}\left[\mathbf{S}\left(\nabla^2_{\mathbf{S}} f(x^k)\right)^{-1}\mathbf{S}^\top\right]\nabla f(x^k). \qquad (29)
\end{aligned}$$

If further $f(x^0)-f(x^*) \leq \varphi^2 \frac{\mu\left(\lambda_{\min}\nabla^2_{\mathbf{S}} f(x^*)\right)^2}{2M^2_{\mathbf{S}}}$ for some $\varphi>0$ we get

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{2}\|\mathbf{S}^\top \nabla f(x^k)\|^2_{(\mathbf{H}(x^k))^{-1}}\right] &\overset{(29)}{\geq} \frac{\nabla f(x^k)^\top \mathbb{E}\left[\mathbf{S}\left(\nabla^2_{\mathbf{S}} f(x^k)\right)^{-1}\mathbf{S}^\top\right]\nabla f(x^k)}{2(1+\varrho)} \\
&\overset{(24)}{\geq} \frac{\nabla f(x^k)^\top \mathbb{E}\left[\mathbf{S}\left(\nabla^2_{\mathbf{S}} f(x^*)\right)^{-1}\mathbf{S}^\top\right]\nabla f(x^k)}{2(1+\varrho)(1+\varphi)} \\
&\overset{(13)}{\geq} \frac{\nabla f(x^k)^\top \left(\zeta\left(\nabla^2 f(x^*)\right)^{-1}\right)\nabla f(x^k)}{2(1+\varrho)(1+\varphi)} \\
&\overset{(24)}{\geq} \frac{(1-\varphi)\zeta\lambda_f(x^k)^2}{2(1+\varrho)(1+\varphi)} \qquad (30)
\end{aligned}$$

Lastly, if $f(x^0)-f(x^*) \leq \omega^{-1}\left(\frac{2\mu^{\frac{3}{2}}}{(1+\gamma^{-1})M}\right)$ where $\omega(y) := y-\log(1+y)$ and $\gamma>0$, we get

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{2}\|\mathbf{S}^\top \nabla f(x^k)\|^2_{(\mathbf{H}(x^k))^{-1}}\right] &\overset{(30)}{\geq} \frac{(1-\varphi)\zeta\lambda_f(x^k)^2}{2(1+\varrho)(1+\varphi)} \\
&\overset{(25)}{\geq} \frac{(1-\varphi)\zeta(f(x^k)-f(x^*))}{(1+\varrho)(1+\varphi)(1+\gamma)}
\end{aligned}$$

and thus (14) follows. In particular for any $\varrho, \gamma > 0, 1 > \varphi > 0$, we can choose

$$\delta = \min\left\{\varrho^4 \frac{2\left(\min_{x\in\chi^0} \lambda_{\min}\nabla_S^2 f(x)\right)^4}{LM_S^2}, \varphi^2 \frac{\mu\left(\lambda_{\min}\nabla_S^2 f(x^*)\right)^2}{2M_S^2}, \omega^{-1}\left(\frac{2\mu^{\frac{3}{2}}}{(1+\gamma^{-1})M}\right)\right\}$$

and

$$\epsilon = 1 - \frac{1-\varphi}{(1+\varrho)(1+\varphi)(1+\gamma)}.$$

□