Unified Convergence Theory of Stochastic and Variance-Reduced Cubic Newton Methods

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study stochastic Cubic Newton methods for solving general possibly non-convex minimization problems. We propose a new framework, which we call the *helper framework*, that provides a unified view of the stochastic and variance-reduced second-order algorithms equipped with global complexity guarantees. It can also be applied to learning with auxiliary information. Our helper framework offers the algorithm designer high flexibility for constructing and analysis of the stochastic Cubic Newton methods, allowing arbitrary size batches, and the use of noisy and possibly biased estimates of the gradients and Hessians, incorporating both the variance reduction and the lazy Hessian updates. We recover the best-known complexities for the stochastic and variance-reduced Cubic Newton, under weak assumptions on the noise and avoiding artificial logarithms. A direct consequence of our theory is the new lazy stochastic second-order method, which significantly improves the arithmetic complexity for large dimension problems. We also establish complexity bounds for the classes of gradient-dominated objectives, that include convex and strongly convex problems. For Auxiliary Learning, we show that using a helper (auxiliary function) can outperform training alone if a given similarity measure is small.

1 Introduction

In many fields of machine learning, it is common to optimize a function $f(\boldsymbol{x})$ that can be expressed as a finite sum:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d}\Big\{ f(\boldsymbol{x}) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}) \Big\}, \tag{1}$$

or, more generally, as an expectation over some given probability distribution: $f(\boldsymbol{x}) = \mathbb{E}_{\zeta}\big[f(\boldsymbol{x},\zeta)\big]$. When f is non-convex, this problem is especially difficult, since finding a global minimum is NP-hard in general [15]. Hence, the reasonable goal is to look for approximate solutions. The most prominent family of algorithms for solving large-scale problems of the form (1) are the *first-order methods*, such as the Stochastic Gradient Descent (SGD) [26, 17]. They employ only stochastic gradient information about the objective $f(\boldsymbol{x})$ and guarantee the convergence to a stationary point, which is a point with a small gradient norm.

Nevertheless, when the objective function is non-convex, a stationary point may be a saddle point or even a local maximum, which is not desirable. Another common issue is that first-order methods typically have a slow convergence rate, particularly when the problem is *ill-conditioned*. Therefore, they may not be suitable when high precision for the solution is required.

To address these challenges, we can take into account *second-order information* (the Hessian matrix) and apply Newton's method (see, e.g. [20]). Among the many versions of this algorithm, the Cubic Newton method [21] is one of the most theoretically established. With the Cubic Newton method, we

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

can guarantee *global convergence* to an approximate *second-order* stationary point (in contrast, the pure Newton method without regularization can even diverge when it starts far from a neighborhood of the solution). For a comprehensive historical overview of the different variants of Newton's method, see [25]. Additionally, the rate of convergence of the Cubic Newton is *provably better* than those for the first-order methods.

Therefore, theoretical guarantees of the Cubic Newton method seem to be very appealing for practical applications. However, the basic version of the Cubic Newton requires the exact gradient and Hessian information in each step, which can be very expensive to compute in the large scale setting. To overcome this issue, several techniques have been proposed:

- One popular approach is to use inexact *stochastic gradient and Hessian estimates* with sub-sampling [34, 18, 33, 22, 13, 6, 1]. This technique avoids using the full oracle information, but typically it has a slower convergence rate compared to the exact Cubic Newton.
- *Variance reduction* techniques [36, 30] combine the advantages of stochastic and exact methods, achievieng an improved rates by recomputing the full gradient and Hessian information at some iterations.
- *Lazy Hessian* updates [27, 10] utilize a simple idea of reusing an old Hessian for several iterations of a second-order scheme. Indeed, since the cost of computing one Hessian is usually much more expensive than one gradient, it can improve the arithmetic complexity of our methods.
- In addition, exploiting the special structure of the function f (if known) can also be helpful. For instance, some studies [21, 19] consider *gradient-dominated objectives*, a subclass of non-convex functions that have improved convergence rates and can even be shown to converge to the global minimum. Examples of such objectives include convex and star-convex functions, uniformly convex functions, and functions satisfying the PL condition [24] as a special case.

In this work, we revise the current state-of-the-art convergence theory for the stochastic Cubic Newton method and propose a unified and improved complexity guarantees for different versions of the method, which combine all the advanced techniques listed above.

Our developments are based on the new *helper framework* for the second-order optimization, that we present in Section 3. For the first-order optimization, a similar in-spirit techniques called *learning with auxiliary information* was developed recently in [8, 31]. Thus, our results can also be seen as a generalization of the Auxiliary Learning paradigm to the second-order optimization. However, note that in our second-order case, we have more freedom for choosing the "helper functions" (namely, we use one for the gradients and one for the Hessians). That brings more flexibility into our methods and it allows, for example, to use the lazy Hessian updates.

Our new helper framework provides us with a unified view of the stochastic and variance-reduced methods and can be used by an algorithm designed to construct new methods. Thus, we show how to recover already known versions of the stochastic Cubic Newton with the best convergence rates, as well as present the new *Lazy Stochastic Second-Order Method*, which significantly improves the total arithmetic complexity for large-dimension problems.

Contributions.

- We introduce the *helper framework* which we argue encompasses multiple methods in a unified way. Such methods include stochastic methods, variance reduction, Lazy methods, core sets, and semi-supervised learning.
- This framework covers previous versions of the variance-reduced stochastic Cubic Newton methods with known rates. Moreover, it provides us with new algorithms that employ *Lazy Hessian* updates and significantly improves the arithmetic complexity (for high dimensions), by using the same Hessian snapshot for several steps of the method.
- In the case of Auxiliary learning we provably show a benefit from using auxiliary tasks as helpers in our framework. In particular, we can replace the smoothness constant by a similarity constant which might be smaller.
- Moreover, our analysis works both for the general class of non-convex functions, as well as for the class of gradient-dominated problems, that includes convex and uniformly convex functions. Hence, in particular, we are the first to establish the convergence rates of the

stochastic Cubic Newton algorithms with variance reduction for the gradient-dominated case.

2 Notation and Assumptions

For simplicity, we consider the finite-sum optimization problem (1), while it can be also possible to generalize our results to arbitrary expectations. We assume that our objective f is bounded from below and denote $f^\star := \inf_{\boldsymbol{x}} f(\boldsymbol{x})$, and use the following notation: $F_0 := f(\boldsymbol{x}_0) - f^\star$, for some initial $\boldsymbol{x}_0 \in \mathbb{R}^d$. We denote by $\|\boldsymbol{x}\| := \langle \boldsymbol{x}, \boldsymbol{x}\rangle^{1/2}$, $\boldsymbol{x} \in \mathbb{R}^d$, the standard Euclidean norm for vectors, and the spectral norm for symmetric matrices, $\|\boldsymbol{H}\| := \max\{\lambda_{\max}(\boldsymbol{H}), -\lambda_{\min}(\boldsymbol{H})\}$, where $\boldsymbol{H} = \boldsymbol{H}^\top \in \mathbb{R}^{d\times d}$. We will also use $x \wedge y$ to denote $\min(x, y)$.

Throughout this work, we make the following smothness assumption on the objective f :

Assumption 1 (Lipschitz Hessian) *The Hessian of f is Lipschitz continuous, for some $L > 0$:*

$$\|\nabla^2 f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{y})\| \quad \leq \quad L\|\boldsymbol{x} - \boldsymbol{y}\|, \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$$

Our goal is to explore the potential of using the Cubically regularized Newton methods to solve problem (1). At each iteration, being at a point $\boldsymbol{x} \in \mathbb{R}^d$, we compute the next point $\boldsymbol{x}^+$ by solving the subproblem of the form

$$\boldsymbol{x}^+ \quad \in \quad \arg\min_{\boldsymbol{y}\in\mathbb{R}^d} \Big\{ \Omega_{M,\boldsymbol{g},\boldsymbol{H}}(\boldsymbol{y}, \boldsymbol{x}) := \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x}\rangle + \tfrac{1}{2}\langle \boldsymbol{H}(\boldsymbol{y} - \boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \tfrac{M}{6}\|\boldsymbol{y} - \boldsymbol{x}\|^3 \Big\}. \tag{2}$$

Here, $\boldsymbol{g}$ and $\boldsymbol{H}$ are estimates of the gradient $\nabla f(\boldsymbol{x})$ and the Hessian $\nabla^2 f(\boldsymbol{x})$, respectively. Note that solving (2) can be done efficiently even for non-convex problems (see [9, 21, 5]). Generally, the cost of computing $\boldsymbol{x}^+$ is $\mathcal{O}(d^3)$ arithmetic operations, which are needed for evaluating an appropriate factorization of $\boldsymbol{H}$. Hence, it is of a similar order as the cost of the classical Newton's step.

We will be interested to find a second-order stationary point to (1). We call (ε, c)-*approximate second-order local minimum* a point $\boldsymbol{x}$ that satisfies:

$$\|\nabla f(\boldsymbol{x})\| \quad \leq \quad \varepsilon \qquad \text{and} \qquad \lambda_{min}(\nabla^2 f(\boldsymbol{x})) \quad \geq \quad -c\sqrt{\varepsilon},$$

where $\varepsilon, c > 0$ are given tolerance parameters. Let us define the following accuracy measure (see [21]):

$$\mu_c(\boldsymbol{x}) \quad := \quad \max\Big(\|\nabla f(\boldsymbol{x})\|^{3/2},\ \tfrac{-\lambda_{min}(\nabla^2 f(\boldsymbol{x}))^3}{c^{3/2}}\Big), \qquad \boldsymbol{x} \in \mathbb{R}^d,\ c > 0.$$

Note that this definition implies that if $\mu_c(\boldsymbol{x}) \leq \varepsilon^{3/2}$ then $\boldsymbol{x}$ is an (ε, c)-approximate local minimum.

Computing gradients and Hessians. It is clear that computing the Hessian matrix can be much more expensive than computing the gradient vector. We denote the corresponding arithmetic complexities by *HessCost* and *GradCost*. We will make and follow the convention that *HessCost* $= d \times$ *GradCost*, where d is the dimension of the problem. For example, this is known to hold for neural networks using the backpropagation algorithm [16]. However, if the Hessian has a sparse structure, the cost of computing the Hessian can be cheaper [23]. Then, we can replace d with the *effective dimension* $d_{\text{eff}} := \frac{\textit{HessCost}}{\textit{GradCost}} \leq d$.

3 Second-Order Optimization with Helper Functions

In this section, we extend the helper framework previously introduced in [8] for first-order optimization methods to second-order optimization.

General principle. The general idea is the following: imagine that, besides the objective function f we have access to a help function h that we think is similar in some sense (that will be defined later) to f and thus it should help to minimize it.

Note that many optimization algorithms can be framed in the following sequential way. For a current state $\boldsymbol{x}$, we compute the next state $\boldsymbol{x}^+$ as:

$$\boldsymbol{x}^+ \quad \in \quad \arg\min_{\boldsymbol{y}\in\mathbb{R}^d}\Big\{ \hat{f}_{\boldsymbol{x}}(\boldsymbol{y}) + M r_{\boldsymbol{x}}(\boldsymbol{y}) \Big\},$$

where $\hat{f}_{\boldsymbol{x}}(\cdot)$ is an approximation of f around current point $\boldsymbol{x}$, and $r_{\boldsymbol{x}}(\boldsymbol{y})$ is a regularizer that encodes how accurate the approximation is, and $M > 0$ is a regularization parameter. In this work, we are interested in cubically regularized second-order models of the form (2) and we use $r_{\boldsymbol{x}}(\boldsymbol{y}) := \frac{1}{6}\|\boldsymbol{y} - \boldsymbol{x}\|^3$.

Now let us look at how we can use a helper h to construct the approximation $\hat{f}$. We notice that we can write

$$f(\boldsymbol{y}) \quad := \quad \underbrace{h(\boldsymbol{y})}_{\text{cheap}} + \underbrace{f(\boldsymbol{y}) - h(\boldsymbol{y})}_{\text{expensive}}$$

We discuss the actual practical choices of the helper function h below. We assume now that we can afford the second-order approximation for the cheap part h around the current point $\boldsymbol{x}$. However, approximating the part $f - h$ can be expensive (as for example when the number of elements n in finite sum (1) is huge), or even impossible (due to lack of data). Thus, we would prefer to approximate the expensive part less frequently. For this reason, let us introduce an extra *snapshot point* $\hat{\boldsymbol{x}}$ that is updated less often than $\boldsymbol{x}$. Then, we use it to approximate $f - h$. Another question that we still need to ask is *what order should we use for the approximation of $f - h$?* We will see that order 0 (approximating by a constant) leads as to the basic stochastic methods, while for orders 1 and 2 we equip our methods with the variance reduction.

Combining the two approximations for h and $f - h$ we get the following model of our objective f:

$$\hat{f}_{\boldsymbol{x},\tilde{\boldsymbol{x}}}(\boldsymbol{y}) \quad = \quad C(\boldsymbol{x},\tilde{\boldsymbol{x}}) + \langle \mathcal{G}(h,\boldsymbol{x},\tilde{\boldsymbol{x}}), \boldsymbol{y} - \boldsymbol{x}\rangle + \tfrac{1}{2}\langle \mathcal{H}(h,\boldsymbol{x},\tilde{\boldsymbol{x}})(\boldsymbol{y} - \boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle, \tag{3}$$

where $C(\boldsymbol{x},\tilde{\boldsymbol{x}})$ is a constant, $\mathcal{G}(h,\boldsymbol{x},\tilde{\boldsymbol{x}})$ is a linear term, and $\mathcal{H}(h,\boldsymbol{x},\tilde{\boldsymbol{x}})$ is a matrix. Note that if $\tilde{\boldsymbol{x}} \equiv \boldsymbol{x}$, then the best second-order model of the form (3) is the Taylor polynomial of degree two for f around $\boldsymbol{x}$, and that would give us the exact Newton-type method. However, when the points $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ are different, we obtain much more freedom in constructing our models.

For using this model in our cubically regularized method (2), we only need to define the gradient $\boldsymbol{g} = \mathcal{G}(h,\boldsymbol{x},\tilde{\boldsymbol{x}})$ and the Hessian estimates $\boldsymbol{H} = \mathcal{H}(h,\boldsymbol{x},\tilde{\boldsymbol{x}})$, and we can also treat them differently (using two different helpers h_1 and h_2, correspondingly). Thus we come to the following general second-order (meta)algorithm. We perform S rounds, the length of each round is $m \geq 1$, which is our key parameter:

Algorithm 1 Cubic Newton with helper functions

Input: $\boldsymbol{x}_0 \in \mathbb{R}^d$, S, $m \geq 1$, $M > 0$.
for $t = 0, \ldots, Sm - 1$ **do**
  if $t \bmod m = 0$ **then**
    Update $\tilde{\boldsymbol{x}}_t$ (using previous states $\boldsymbol{x}_{i\leq t}$)
  else
    $\tilde{\boldsymbol{x}}_t = \tilde{\boldsymbol{x}}_{t-1}$
  Form helper functions h_1, h_2
  Compute the gradient $\boldsymbol{g}_t = \mathcal{G}(h_1,\boldsymbol{x}_t,\tilde{\boldsymbol{x}}_t)$, and the Hessian $\boldsymbol{H}_t = \mathcal{H}(h_2,\boldsymbol{x}_t,\tilde{\boldsymbol{x}}_t)$
  Compute the cubic step $\boldsymbol{x}_{t+1} \in \arg\min_{\boldsymbol{y}\in\mathbb{R}^d} \Omega_{M,\boldsymbol{g}_t,\boldsymbol{H}_t}(\boldsymbol{y},\boldsymbol{x}_t)$
return $\boldsymbol{x}_{out}$ using the history $(\boldsymbol{x}_i)_{0\leq i\leq Sm}$

In Algorithm 1 we update the snapshot $\tilde{\boldsymbol{x}}$ regularly every m iterations. The two possible options are

$$\tilde{\boldsymbol{x}}_t \quad = \quad \boldsymbol{x}_{t \bmod m} \qquad \text{(use the last iterate)} \tag{4}$$

or

$$\tilde{\boldsymbol{x}}_t \quad = \quad \arg\min_{i\in\{t-m+1,\ldots,t\}} f(\boldsymbol{x}_i) \qquad \text{(use the best iterate)} \tag{5}$$

Clearly, option (5) is available only in case we can efficiently estimate the function values. However, we will see that it serves us with better global convergence guarantees, for the gradient-dominated functions.

It remains only to specify how we choose the helpers h_1 and h_2. We need to assume that they are somehow similar to f. Let us present several efficient choices that lead to implementable second-order schemes.

3.1 Basic Stochastic Methods

If the objective function f is very "expensive" (for example of the form (1) with $n \to \infty$), one option is to ignore the part $f - h$ i.e. to approximate it by a zeroth-order approximation: $f(\boldsymbol{y}) - h(\boldsymbol{y}) \approx f(\tilde{\boldsymbol{x}}) - h(\tilde{\boldsymbol{x}})$. Since it is just a constant, we do not need to update $\tilde{\boldsymbol{x}}$. In this case, we have:

$$\mathcal{G}(h_1, \boldsymbol{x}, \tilde{\boldsymbol{x}}) \;:=\; \nabla h_1(\boldsymbol{x}), \qquad \mathcal{H}(h_2, \boldsymbol{x}, \tilde{\boldsymbol{x}}) \;:=\; \nabla^2 h_2(\boldsymbol{x})\,.$$

To treat this choice of the helpers and motivated by the form of the errors in Lemma 5, we assume the following similarity assumptions:

Assumption 2 (Bounded similarity) *Let for some $\delta_1, \delta_2 \geq 0$, it holds*

$$\mathbb{E}_{h_1}[\|\mathcal{G}(h_1, \boldsymbol{x}, \tilde{\boldsymbol{x}}) - \nabla f(\boldsymbol{x})\|^{3/2}] \leq \delta_1^{3/2},\ \mathbb{E}_{h_2}[\|\mathcal{H}(h_2, \boldsymbol{x}, \tilde{\boldsymbol{x}}) - \nabla^2 f(\boldsymbol{x})\|^3] \leq \delta_2^3, \quad \forall \boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathbb{R}^d.$$

Under this assumption, we prove the following theorem:

Theorem 1 *Under Assumptions 1 and 2, and $M \geq L$, for an output of Algorithm 1 $\boldsymbol{x}_{out}$ chosen uniformly at random from $(\boldsymbol{x}_i)_{0 \leq i \leq Sm}$, we have:*

$$\mathbb{E}[\mu_M(\boldsymbol{x}_{out})] \;\;=\;\; \mathcal{O}\Big(\tfrac{\sqrt{M}F_0}{Sm} + \tfrac{\delta_2^3}{M^{3/2}} + \delta_1^{3/2}\Big).$$

We see that according to this result, we can get $\mathbb{E}[\mu_M(\boldsymbol{x}_{out})] \leq \varepsilon^{3/2}$ only for $\varepsilon > \delta_1$. In other words, we can converge only to a certain *neighbourhood around a stationary point*, that is determined by the error δ_1 of the stochastic gradients.

However, as we will show next, this seemingly pessimistic dependence leads to the same rate of classical subsampled Cubic Newton methods discovered in [18, 33, 34].

Let us discuss now the specific case of stochastic optimization, where f has the specific form (1), with n potentially being very large. In this case, it is customary to sample batches at random and assume the noise to be bounded in expectation. Precisely speaking, if we assume the standard assumption that for one index sampled uniformly at random, we have $\mathbb{E}_i\|\nabla f(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x})\|^2 \leq \sigma_g^2$ and $\mathbb{E}_i\|\nabla^2 f(\boldsymbol{x}) - \nabla^2 f_i(\boldsymbol{x})\|^3 \leq \sigma_h^3$, then it is possible to show that for

$$h_1 \;\;=\;\; \tfrac{1}{b_g}\textstyle\sum_{i \in \mathcal{B}_g} f_i \qquad \text{and} \qquad h_2 \;=\; \tfrac{1}{b_h}\textstyle\sum_{i \in \mathcal{B}_h} f_i, \tag{6}$$

for batches $\mathcal{B}_g, \mathcal{B}_h \subseteq [n]$ sampled uniformly at random and of sizes b_g and b_h respectively, Assumption 2 is satisfied with [28]: $\delta_1 = \frac{\sigma_g}{\sqrt{b_g}}$ and $\delta_2 = \tilde{\mathcal{O}}(\frac{\sigma_h}{\sqrt{b_h}})$. Note that we can use the same random subsets of indices $\mathcal{B}_g, \mathcal{B}_h$ for all iterations.

Corollary 1 *In Algorithm 1, let us choose $M = L$ and $m = 1$, with basic helpers* (6)*. Then, according to Theorem 1, for any $\varepsilon > 0$, to reach an (ε, L)-approximate second-order local minimum, we need at most $S = \frac{\sqrt{L}F_0}{\varepsilon^{3/2}}$ iterations with $b_g = \left(\frac{\sigma_g}{\varepsilon}\right)^2$ and $b_h = \frac{\sigma_h^2}{\varepsilon}$. Therefore, the total arithmetic complexity of the method becomes*

$$\mathcal{O}\Big(\tfrac{\sigma_g^2}{\varepsilon^{7/2}} + \tfrac{\sigma_h^2}{\varepsilon^{5/2}} d_{\mathit{eff}}\Big) \times \mathit{GradCost}.$$

It improves upon the complexity $\mathcal{O}(\frac{1}{\varepsilon^4}) \times \mathit{GradCost}$ of the first-order SGD for non-convex optimization [12], unless $d_{\text{eff}} > \frac{1}{\varepsilon^{3/2}}$ (high cost of computing the Hessians).

3.2 Let the Objective Guide Us

If the objective f is such that we can afford to access its gradients and Hessians from time to time (functions of the form (1) with $n < \infty$ and "reasonable"), then we can do better than the previous chapter. In this case, we can afford to use a better approximation of the term $f(\boldsymbol{y}) - h(\boldsymbol{y})$. From a theoretical point of view, we can treat the case when f is only differentiable once, and thus we can only use a first-order approximation of $f - h$, in this case, we will only be using the hessian of the helper h but only gradients of f. However, in our case, if we assume we have access to gradients then

we can also have access to the Hessians of f as well (from time to time). For this reason, we consider a second-order approximation of the term $f - h$, if we follow the procedure that we described above we find:

$$\mathcal{G}(h_1, \boldsymbol{x}, \tilde{\boldsymbol{x}}) := \nabla h_1(\boldsymbol{x}) - \nabla h_1(\tilde{\boldsymbol{x}}) + \nabla f(\tilde{\boldsymbol{x}}) + (\nabla^2 f(\tilde{\boldsymbol{x}}) - \nabla^2 h_1(\tilde{\boldsymbol{x}}))(\boldsymbol{x} - \tilde{\boldsymbol{x}}) \quad (7)$$

$$\mathcal{H}(h_2, \boldsymbol{x}, \tilde{\boldsymbol{x}}) := \nabla^2 h_2(\boldsymbol{x}) - \nabla^2 h_2(\tilde{\boldsymbol{x}}) + \nabla^2 f(\tilde{\boldsymbol{x}}) \quad (8)$$

We see that there is an explicit dependence on the snapshot $\tilde{\boldsymbol{x}}$ and thus we need to address the question of how this snapshot point should be updated in Algorithm 1. In general, we can update it with a certain probability $p \sim \frac{1}{m}$, and we can use more advanced combinations of past iterates (like the average). However, for our purposes, we simply choose option 4 (i.e. the last iterate), thus it is only updated once every m iterations.

We also need to address the question of the measure of similarity in this case. Since we are using a second-order approximation of $f - h$, it is very logical to compare them using the difference between their third derivatives or equivalently, the Hessian Lipschitz constant of their difference. Precisely we make the following similarity assumption :

Assumption 3 (Lipschitz similarity) *Let for some $\delta_1, \delta_2 \geq 0$, it holds, $\forall \boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathbb{R}^d$:*

$$\mathbb{E}_{h_1}[\|\mathcal{G}(h_1, \boldsymbol{x}, \tilde{\boldsymbol{x}}) - \nabla f(\boldsymbol{x})\|^{3/2}] \leq \delta_1^{3/2} \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^3,$$

$$\mathbb{E}_{h_2}[\|\mathcal{H}(h_2, \boldsymbol{x}, \tilde{\boldsymbol{x}}) - \nabla^2 f(\boldsymbol{x})\|^3] \leq \delta_2^3 \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^3.$$

In particular, if $f - h_1$ and $f - h_2$ have δ_1 and δ_2 Lipschitz Hessians respectively then h_1 and h_2 satisfy Assumption 3.

Under this assumption, we show that the errors resulting from the use of the snapshot can be successfully balanced by choosing M satisfying:

$$4\big(\tfrac{\delta_1}{M}\big)^{3/2} + 73\big(\tfrac{\delta_2}{M}\big)^3 \leq \tfrac{1}{24m^3}. \quad (9)$$

And we have the following theorem.

Theorem 2 *For f, h_1, h_2 verifying Assumptions 1,3. For a regularization parameter M chosen such that $M \geq L$ and (9) is satisfied. For an output of Algorithm 1 $\boldsymbol{x}_{out}$ chosen uniformly at random from $(\boldsymbol{x}_i)_{0 \leq i \leq Sm:=T}$, we have:*

$$\mathbb{E}[\mu_M(\boldsymbol{x}_{out})] = \mathcal{O}\Big(\tfrac{\sqrt{M}F_0}{Sm}\Big),$$

In particular, we can choose $M = \max(L, 32\delta_1 m^2, 16\delta_2 m)$ which gives

$$\mathbb{E}[\mu_M(\boldsymbol{x}_{out})] = \mathcal{O}\Big(\tfrac{\sqrt{L}F_0}{Sm} + \tfrac{\sqrt{\delta_2}F_0}{S\sqrt{m}} + \tfrac{\sqrt{\delta_1}F_0}{S}\Big). \quad (10)$$

Based on the choices of the helpers h_1 and h_2 we can have many algorithms. We discuss these in the following sections. We start by discussing variance reduction and Lazy Hessians which rely on sampling batches randomly, then move to core-sets which try to find, more intelligently, representative weighted batches of data, after this, we discuss semi-supervised learning and how unlabeled data can be used to engineer the helpers. More generally, auxiliary learning tries to leverage auxiliary tasks in training a given main task, the auxiliary tasks can be treated as helpers.

3.3 Variance Reduction and Lazy Hessians

The following lemma demonstrates that we can create helper functions h with lower similarity to the main function f of the form (1) by employing sampling and averaging.

Lemma 1 *Let $f = \frac{1}{n}\sum_{i=1}^n f_i$ such that all f_i are twice differentiable and have L-Lipschitz Hessians. Let $\mathcal{B} \subset \{1, \cdots, n\}$ be of size b and sampled with replacement uniformly at random, and define $h_{\mathcal{B}} = \frac{1}{b}\sum_{i\in\mathcal{B}} f_i$, then $h_{\mathcal{B}}$ satisfies Assumption 3 with $\delta_1 = \frac{L}{\sqrt{b}}$ and $\delta_2 = \mathcal{O}(\frac{\sqrt{\log(d)}L}{\sqrt{b}})$.*

Choice of the parameter m in Algorithm 1. Minimizing the total arithmetic cost, we choose $m = \arg\min_m \#Grad(m,\varepsilon) + d\#Hess(m,\varepsilon)$, where $\#Grad(m,\varepsilon)$ and $\#Hess(m,\varepsilon)$ denote the number of gradients and Hessians required to find an ε stationary point.

Now we are ready to discuss several special cases that are direct consequences from Theorem 2.

First, note that choosing $h_1 = h_2 = f$ gives the classical Cubic Newton method [21], whereas choosing $h_1 = f$ and $h_2 = 0$, gives the Lazy Cubic Newton [10]. In both cases, we recuperate the known rates of convergence.

General variance reduction. If we sample batches $\mathcal{B}_g$ and $\mathcal{B}_h$ of sizes b_g and b_h consecutively at random and choose

$$h_1 \;=\; \tfrac{1}{b_g}\textstyle\sum_{i\in\mathcal{B}_g} f_i \qquad \text{and} \qquad h_2 \;=\; \tfrac{1}{b_h}\textstyle\sum_{i\in\mathcal{B}_h} f_i,$$

and use these helpers along with the estimates (7), (8), we obtain the *Variance Reduced Cubic Newton* algorithm [36, 30]. According to Lemma 1, this choice corresponds to $\delta_1 = \frac{L}{\sqrt{b_g}}$ and $\delta_2 = \tilde{\mathcal{O}}(\frac{L}{\sqrt{b_h}})$. For $b_g \sim m^4 \wedge n, b_h \sim m^2 \wedge n$ and $M = L$, we have the non-convex convergence rate $\mathcal{O}\big(\frac{\sqrt{L}F_0}{Sm}\big)$, which is the same as that of the cubic Newton algorithm but with a smaller cost per iteration. Minimizing the total arithmetic cost, we can choose $m = \arg\min_m \frac{dn+d(m^3\wedge nm)+(m^5\wedge nm)}{m}$. Let us denote by $g^{VR}(n,d)$ the corresponding optimal value. Then we reach an (ε, L)-approximate second-order local minimum in at most $\mathcal{O}(\frac{g^{VR}(n,d)}{\varepsilon^{3/2}}) \times$ *GradCost* arithmetic operations.

Variance reduction with Lazy Hessians. We can also use lazy updates for Hessians combined with variance-reduced gradients. This corresponds to choosing

$$h_1 \;=\; \tfrac{1}{b_g}\textstyle\sum_{i\in\mathcal{B}_g} f_i \qquad \text{and} \qquad h_2 \;=\; 0,$$

which implies (according to Lemma 1) that $\delta_1 = \frac{L}{\sqrt{b_g}}$ and $\delta_2 = L$. In this case, we need $b_g \sim m^2$ to obtain a convergence rate of $\mathcal{O}\big(\frac{\sqrt{L}F_0}{S\sqrt{m}}\big)$, which matches the convergence rate of the Lazy Cubic Newton method while using stochastic gradients. We choose this time $m = \arg\min_m \frac{nd+(m^3\wedge mn)}{\sqrt{m}}$, as before. Let us denote $g^{Lazy}(n,d)$ the corresponding minimum. Then we guarantee to reach an (ε, mL)-approximate second-order local minimum in at most $\mathcal{O}(\frac{g^{Lazy}(n,d)}{\varepsilon^{3/2}}) \times$ *GradCost* operations.

To be lazy or not to be? We show that $g^{Lazy}(n,d) \sim (nd)^{5/6} \wedge n\sqrt{d}$ and $g^{VR}(n,d) \sim (nd)^{4/5} \wedge (n^{2/3}d + n)$. In particular, for $d \geq n^{2/3}$ we have $g^{Lazy}(n,d) \leq g^{VR}(n,d)$ and thus for $d \geq n^{2/3}$ *it is better to use Lazy Hessians* than variance-reduced Hessians from a gradient equivalent cost perspective. We note also that for the Lazy approach, we can keep a factorization of the Hessian (this factorization induces most of the cost of solving the cubic subproblem) and thus it is as if we only need to solve the subproblem once every m iterations, so the Lazy approach has a big advantage compared to the general approach, and the advantage becomes even bigger for the case of large dimensions.
Note that according to the theory, we could use the same random batches $\mathcal{B}_g, \mathcal{B}_h \subseteq [n]$ generated once for all iterations. However, using the resampled batches can lead to a more stable convergence.

3.4 Other Applications

The result in (10) is general enough that it can include many other applications that are only limited by our imagination. To cite a few such applications there are:

Core sets. [3] The idea of core sets is simple: can we summarize a potentially big data set using only a few (weighted) important examples? Many reasons such as redundancy make the answer yes. Devising approaches to find such core sets is outside of the scope of this work, but in general, we can see from (10) that if we have batches $\mathcal{B}_g, \mathcal{B}_h$ such that they are $(\delta_1, 1)$ and $(\delta_2, 2)$ similar to f respectively, then we can keep reusing the same batch $\mathcal{B}_g$ for at least $\sqrt{\frac{L}{\delta_1}}$ times, and $\mathcal{B}_h$ for $\frac{L}{\delta_2}$ all the while guaranteeing an improved rate. So then if we can design such small batches with small δ_1 and δ_2 then we can keep reusing them, and joy the improved rate without needing large batches.

Auxiliary learning. [4, 2, 32] study how a given task f can be trained in the presence of auxiliary (related) tasks. Our approach can be indeed used for auxiliary learning by treating the auxiliaries as helpers. If we compare (10) to the rate that we obtained without the use of the helpers: $\mathcal{O}(\frac{\sqrt{L}F_0}{S})$, we see that we have a better rate using the helpers/auxiliary tasks when $\frac{1}{m} + \frac{\sqrt{\delta_2}}{\sqrt{mL}} + \frac{\sqrt{\delta_1}}{\sqrt{L}} \leq 1$.

Semi-supervised learning.[35] Semi-supervised learning is a machine learning approach that combines the use of both labeled data and unlabeled data during training. In general, we can use the unlabeled data to construct the helpers, we can start for example by using random labels for the helpers and improving the labels with training. There are at least two special cases where our theory implies improvement by only assigning random labels to the unlabeled data. In fact, for both regularized least squares and logistic regression, we notice that the Hessian is independent of the labels (only depends on inputs) and thus if the unlabeled data comes from the same distribution as the labeled data, then we can use it to construct helpers which, at least theoretically, have $\delta_1 = \delta_2 = 0$. Because the Hessian is independent of the labels, we can technically endow the unlabeled data with random labels. Theorem 2 would imply in this case $\mathbb{E}[\mu_L(\boldsymbol{x}_{out})] = \mathcal{O}(\frac{\sqrt{L}F_0}{Sm})$, where S is the number of times we use labeled data and $S(m-1)$ is the number of unlabeled data.

4 Gradient-Dominated Functions

We consider now the class of gradient-dominated functions defined below.

Assumption 4 (τ, α)***-gradient dominated.*** *A function f is called gradient dominated on set if it holds, for some $\alpha \geq 1$ and $\tau > 0$:*

$$f(\boldsymbol{x}) - f^\star \quad \leq \quad \tau \|\nabla f(\boldsymbol{x})\|^\alpha, \qquad \forall \boldsymbol{x} \in \mathbb{R}^d. \tag{11}$$

Examples of functions satisfying this assumption are convex functions ($\alpha = 1$) and strongly convex functions ($\alpha = 2$), see Appendix D.1. For such functions, we can guarantee convergence (in expectation) to a *global minimum*, i.e. we can find a point $\boldsymbol{x}$ such that $f(\boldsymbol{x}) - f^\star \leq \varepsilon$.

The Gradient-dominance property is interesting because many non-convex functions have been shown to satisfy it [29, 14, 19]. Furthermore, besides convergence to a global minimum, we get accelerated rates.

We note that for $\alpha > 3/2$ (and only for this case), we needed to assume the following (stronger) inequality:

$$\mathbb{E} f(\boldsymbol{x}_t) - f^\star \quad \leq \quad \tau \mathbb{E}\big[\|\nabla f(\boldsymbol{x}_t)\|\big]^\alpha, \tag{12}$$

where the expectation is taken with respect to the iterates $(\boldsymbol{x}_t)$ of our algorithms. This is a stronger assumption than (11). To avoid using this stronger assumption, we can assume that the iterates belong to some compact set $Q \subset \mathbb{R}^d$ and that the gradient norm is uniformly bounded: $\forall \boldsymbol{x} \in Q : \|\nabla f(\boldsymbol{x})\| \leq G$. Then, a (τ, α)-gradient dominated on set Q function is also a $(\tau G^{\alpha - 3/2}, 3/2)$-gradient dominated on this set for any $\alpha > 3/2$.

In Theorem 3 we extend the results of Theorem 1 to gradient-dominated functions.

Theorem 3 *Under Assumptions 1,2,4, for $M \geq L$ and $T := Sm$ we have:*
- For $1 \leq \alpha \leq 3/2$: $\mathbb{E}[f(\boldsymbol{x}_T)] - f^\star = \mathcal{O}\Big(\big(\frac{\alpha\sqrt{M}\tau^{3/(2\alpha)}}{(3-2\alpha)T}\big)^{\frac{2\alpha}{3-2\alpha}} + \tau\frac{\delta_2^{2\alpha}}{M^\alpha} + \tau\delta_1^\alpha \Big)$.

- For $3/2 < \alpha \leq 2$, let $h_0 = \mathcal{O}(\frac{F_0}{(\sqrt{M}\tau^{\frac{3}{2\alpha}})^{\frac{2\alpha}{3-2\alpha}}})$, then for $T \geq t_0 = \mathcal{O}(h_0^{\frac{3-2\alpha}{2\alpha}}\log(h_0))$ we have:

$$E[f(\boldsymbol{x}_T)] - f^\star = \mathcal{O}\Big((\sqrt{M}\tau^{\frac{3}{2\alpha}})^{\frac{2\alpha}{3-2\alpha}} \big(\tfrac{1}{2}\big)^{(\frac{2\alpha}{3})^{T-t_0}} + \tau\frac{\delta_2^{2\alpha}}{M^\alpha} + \tau\delta_1^\alpha \Big) .$$

Theorem 3 shows (up to the noise level) for $1 \leq \alpha < 3/2$ a sublinear rate, for $\alpha = 3/2$ a linear rate (obtained by taking the limit $\alpha \to 3/2$) and a superlinear rate for $\alpha > 3/2$.

We do the same thing for Theorem 2 which we extend in Theorem 4. In this case, we need to set the snapshot line 3 in Algorithm 1) as in 5 i.e. the snapshot corresponds to the state with the smallest value of f during the last m iterations.

Theorem 4 *Under Assumptions 1,3,4, for $M = \max(L, 34\delta_1 m^2, 11\delta_2 m)$, we have:*
- For $1 \leq \alpha \leq 3/2$: $\mathbb{E}[f(\boldsymbol{x}_{Sm})] - f^\star = \mathcal{O}\Big(\big(\frac{\alpha\sqrt{M}\tau^{3/(2\alpha)}}{(3-2\alpha)Sm}\big)^{\frac{2\alpha}{3-2\alpha}} \Big)$.

- For $3/2 < \alpha \leq 2$, let $h_0 = \mathcal{O}(\frac{F_0}{(\frac{\sqrt{M}}{m}\tau^{\frac{3}{2\alpha}})^{\frac{2\alpha}{3-2\alpha}}})$, then for $S \geq s_0 = \mathcal{O}(h_0^{\frac{3-2\alpha}{2\alpha}}\log(h_0))$ we have:

$$\mathbb{E}[f(\boldsymbol{x}_{Sm})] - f^\star = \Big((\tfrac{\sqrt{M}}{m}\tau^{\frac{3}{2\alpha}})^{\frac{2\alpha}{3-2\alpha}} \big(\tfrac{1}{2}\big)^{(\frac{2\alpha}{3})^{S-s_0}} \Big)$$

Again, the same behavior is observed as for Theorem 3 but this time without noise (variance reduction is working). To the best of our knowledge, this is the first time such analysis is made. As a direct consequence of our results, we obtain new global complexities for the variance-reduced and lazy variance-reduced Cubic Newton methods on the class of gradient-dominated functions.

To compare the statements of Theorems 3 and 4, for convex functions (i.e. $\alpha = 1$), Theorem 3 guarantees convergence to a $\varepsilon-$global minimum in at most $\mathcal{O}(\frac{1}{\varepsilon^{5/2}} + \frac{d}{\varepsilon^{3/2}})$ *GradCost*, whereas Theorem 4 only needs $\mathcal{O}(\frac{g(n,d)}{\sqrt{\varepsilon}})$ *GradCost*, where $g(n,d)$ is either $g^{Lazy}(n,d) = (nd)^{5/6} \wedge n\sqrt{d}$ or $g^{VR}(n,d) = (nd)^{4/5} \wedge (n^{2/3}d + n)$. See the Appendix D.3 for more details.

5 Limitations and possible extensions

Estimating similarity between the helpers and the main function. While we show in this work that we can have an improvement over training alone, this supposes that we know the similarity constants δ_1, δ_2, hence it will be interesting to have approaches that can adapt to such constants.
Engineering helper functions. Building helper task with small similarities is also an interesting idea. Besides the examples in supervised learning and core-sets that we provide, it is not evident how to do it in a generalized way.
Using the helper to regularize the cubic subproblem. We note that while we proposed to approximate the "cheap" part as well in Section 3, one other theoretically viable approach is to keep it intact and approximately solve a "proximal type" problem involving h, this will lead to replacing L by δ, but the subproblem is even more difficult to solve. However our theory suggests that we don't need to solve this subproblem exactly, we only need $m \geq \frac{L}{\delta}$. We do not treat this case here.

6 Conclusion

In this work, we proposed a general theory for using auxiliary information in the context of the cubically regularized Newton's method. Our theory encapsulates the classical stochastic methods as well as variance reduction and Lazy methods. For auxiliary learning, we showed a provable benefit compared to training alone. Besides studying the convergence for general non-convex functions for which we show convergence to approximate local minima, we also study gradient-dominated functions, for which convergence is accelerated and is to approximate global minima.

References

[1] A. Agafonov, D. Kamzolov, P. Dvurechensky, A. Gasnikov, and M. Takáč. Inexact tensor methods and their application to stochastic convex optimization. *arXiv preprint arXiv:2012.15636*, 2020.

[2] N. Aviv, A. Idan, M. Haggai, C. Gal, and F. Ethan. Auxiliary learning by implicit differentiation. *ICLR 2021*.

[3] O. Bachem, M. Lucic, and K. Andreas. Practical coreset constructions for machine learning. *arXiv:1703.06476 [stat.ML]*`https://arxiv.org/abs/1703.06476`, 2017.

[4] S. Baifeng, H. Judy, S. Kate, D. Trevor, and X. Huijuan. Auxiliary task reweighting for minimum-data learning. *34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.*

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

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

[7] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[8] E. M. Chayti and S. P. Karimireddy. Optimization with access to auxiliary information. *arXiv:2206.00395 [cs.LG]*, 2022.

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

[10] N. Doikov, E. M. Chayti, and M. Jaggi. Second-order optimization with lazy hessians. *arXiv:2212.00781 [math.OC]*, 2022.

[11] N. Doikov and Y. Nesterov. Minimizing uniformly convex functions by cubic regularization of newton method. *Journal of Optimization Theory and Applications 189:317–339*, 2021.

[12] S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

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

[14] M. Hardt and T. Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

[15] C. J. Hillar and L.-H. Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM) 60 45.*, 2013.

[16] H. J. Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10):947–954, 1960.

[17] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist. Volume 23, Number 3, 462-466*, 1952.

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

[19] S. Masiha, S. Salehkaleybar, N. He, N. Kiyavash, and P. Thiran. Stochastic second-order methods improve best-known sample complexity of sgd for gradient-dominated functions. In *NeurIPS 2022 - Advances in Neural Information Processing Systems*, volume 35, pages 10862–10875, 2022.

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

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

[22] T. Nilesh, S. Mitchell, J. Chi, R. Jeffrey, and J. Michael I. Stochastic cubic regularization for fast nonconvex optimization. *Part of Advances in Neural Information Processing Systems 31*, 2018.

[23] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[24] B. T. Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 3(4):643–653,*, 1963.

[25] B. T. Polyak. Newton's method and its use in optimization. *European Journal of Operational Research*, 181(3):1086–1096, 2007.

[26] H. Robbins and S. Monro. A stochastic approximation method the annals of mathematical statistics. *Vol. 22, No. 3. pp. 400-407*, 1951.

[27] V. Shamanskii. A modification of Newton's method. *Ukrainian Mathematical Journal*, 19(1):118–122, 1967.

[28] J. A. Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[29] L. uanzhi and Y. Yang. Convergence analysis of two-layer neural networks with relu activation. *Advances in neural information processing systems, 30*, 2017.

[30] Z. Wang, Z. Yi, L. Yingbin, and L. Guanghui. Stochastic variance-reduced cubic regularization for nonconvex optimization. *AISTATS*, 2019.

[31] B. Woodworth, K. Mishchenko, and F. Bach. Two losses are better than one: Faster optimization using a cheaper proxy. *arXiv preprint arXiv:2302.03542*, 2023.

[32] L. Xingyu, S. B. Harjatin, K. George, and H. David. Adaptive auxiliary task weighting for reinforcement learning. *33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.*

[33] P. Xu, F. Roosta-Khorasani, and M. W. Mahoney. Newton-type methods for non-convex optimization under inexact Hessian information. *arXiv preprint arXiv:1708.07164* ., 2017.

[34] P. Xu, J. Yang, F. Roosta-Khorasani, and M. W. Mahoney. Sub-sampled newton methods with non-uniform sampling. 2016.

[35] X. Yang, Z. Song, I. King, and Z. Xu. A survey on deep semi-supervised learning. Technical report, 2021.

[36] D. Zhou, P. Xu, and Q. Gu. Stochastic variance-reduced cubic regularization methods. *Journal of Machine Learning Research 20 1-47*, 2019.

0 Experiments

0.1 To be lazy or not

To verify our findings from Subsection 3.3, we consider a logistic regression problem on the “a9a" data set [7]. Specifically

$$f(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} \log(1 + \exp(-b_i \boldsymbol{x}^\top \boldsymbol{a}_i)) + \frac{\lambda}{2}\|\boldsymbol{x}\|^2 \,,$$

where $\{(\boldsymbol{a}_i, b_i)\}_{i=1}^N$ are samples from our data, and $\lambda \geq 0$ is a regularization parameter.

We consider the variance-reduced cubic Newton method from [36] (referred to as “full VR"), its lazy version where we do not update the snapshot Hessian (“Lazy VR"), the stochastic Cubic Newton method (“SCN"), the Cubic Newton algorithm (“CN"), Gradient Descent with line search (“GD") and Stochastic Gradient Descent (“SGD"). We report the results in terms of time and gradient arithmetic computations needed to arrive at a given level of convergence.

Figure 1 shows how the lazy version saves both time and arithmetic computations without sacrificing the convergence precision.

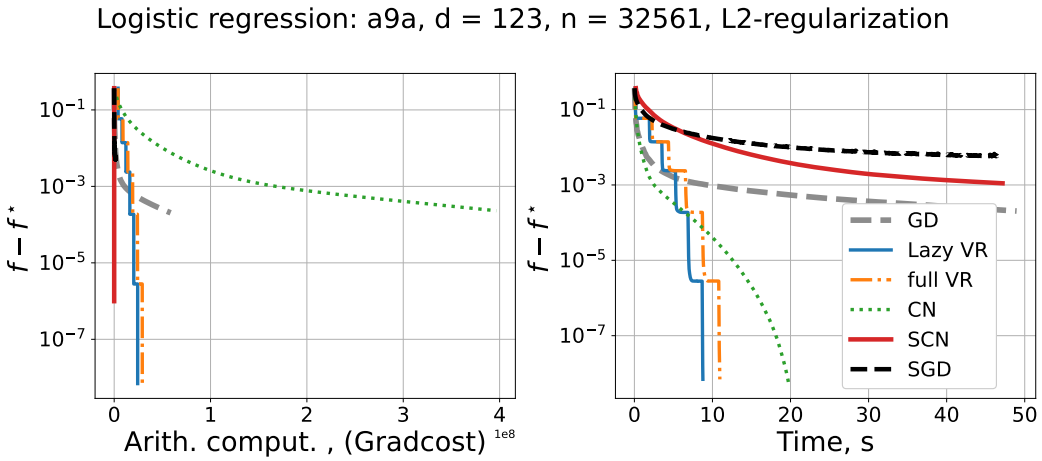

Figure 1: Comparison of the convergence of different algorithms. We see that “Lazy VR" has the same convergence speed as its full version “full VR" and the cubic Newton method “CN" while needing less time and fewer arithmetic computations.

The *Gradcost* is computed using the convention that computing one hessian is d times as expensive as computing one gradient.

0.2 Auxiliary Learning

Our goal is to show that the helper framework is very general and that it goes beyond the variance reduction and lazy Hessian computations. For the previously considered problem of training the logistic regression (using the same “a9a" data set), we suppose that we also have access to unlabeled data (in this sense this becomes semi-supervised learning). Specifically, we have a labeled dataset $\mathcal{D}_l = \{(\boldsymbol{a}_i, b_i)\}_{i=1}^{N_l}$ and an unlabeled data set $\mathcal{D}_u = \{\boldsymbol{a}_i\}_{i=N_l+1}^{N_l+N_u}$, we suppose that both data sets are sampled from the same distribution $\mathcal{P}_{(\boldsymbol{a},b)}$.

Our goal is to minimize

$$f(\boldsymbol{x}) = \mathbb{E}_{(\boldsymbol{a},b)\sim\mathcal{P}_{(\boldsymbol{a},b)}}[\log(1 + \exp(-b\boldsymbol{x}^\top \boldsymbol{a}))] \,.$$

A simple computation shows that the Hessian of f only depends on $\mathcal{P}_{\boldsymbol{a}}$, and, for this reason, we can use unlabeled data to construct a good approximation of the true Hessian (if we can sample from $\mathcal{P}_{\boldsymbol{a}}$, we construct the exact Hessian and thus have a helper h with $\delta_1 = \delta_2 = 0$).

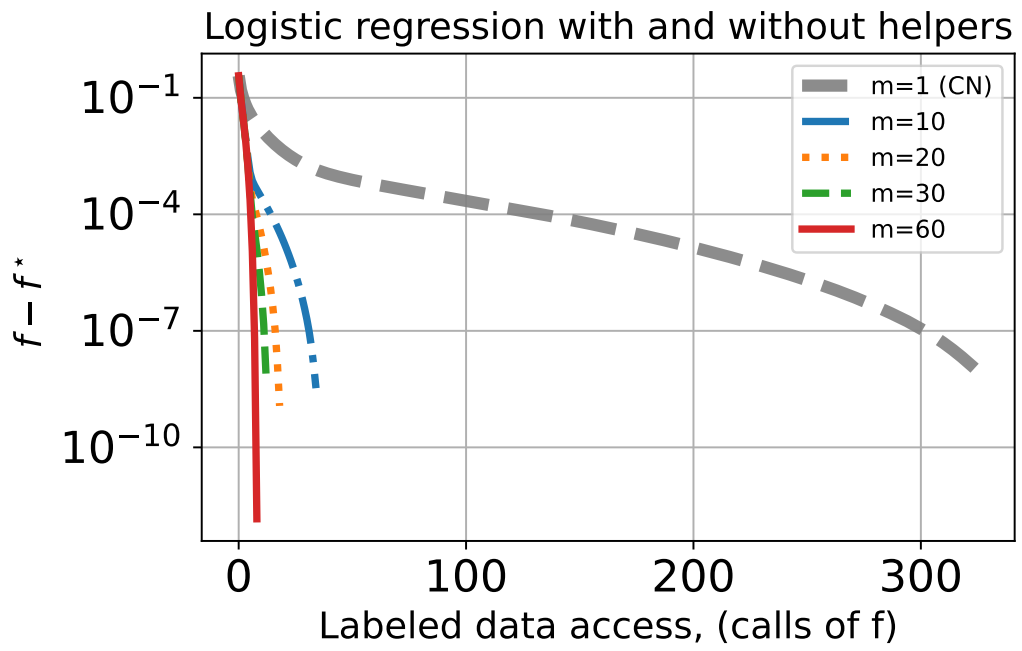

Figure 2: Cubic Newton method with and without using the helper function h. For $m = 1$ this is simply the classic Cubic Newton method. To give an intuitive meaning to the plot, $\frac{1}{m}$ is the percentage of labeled data used during training. We can clearly see that using our approach we benefit a lot from the helper function h.

Let

$$h(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{a}\sim\mathcal{P}_{\boldsymbol{a}},\boldsymbol{b}\sim \mathit{Random}\{\pm 1\}}[\log(1+\exp(-b\boldsymbol{x}^\top \boldsymbol{a}))] \ ,$$

where $\mathit{Random}\{\pm 1\}$ is any distribution on labels. In our experiments, we use uniform distribution. Figure 2 shows that, indeed we can benefit a lot from using this helper function.

0.3 Additional experiments

We go back to comparing the algorithms in 0.1. We consider now non-convex problems.

First we consider logistic regression with a non-convex regularizer $Reg(\boldsymbol{x}) = \sum_{i=1}^{d} \frac{\boldsymbol{x}_i^2}{1+\boldsymbol{x}_i^2}$. Precisely speaking we minimize

$$f(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} \log(1+\exp(-b_i \boldsymbol{x}^\top \boldsymbol{a}_i)) + \lambda Reg(\boldsymbol{x}) \ .$$

Figure 3 shows the results in this case. Again we see that “lazy VR" reduces both time and gradient equivalent computations without sacrificing the convergence speed.

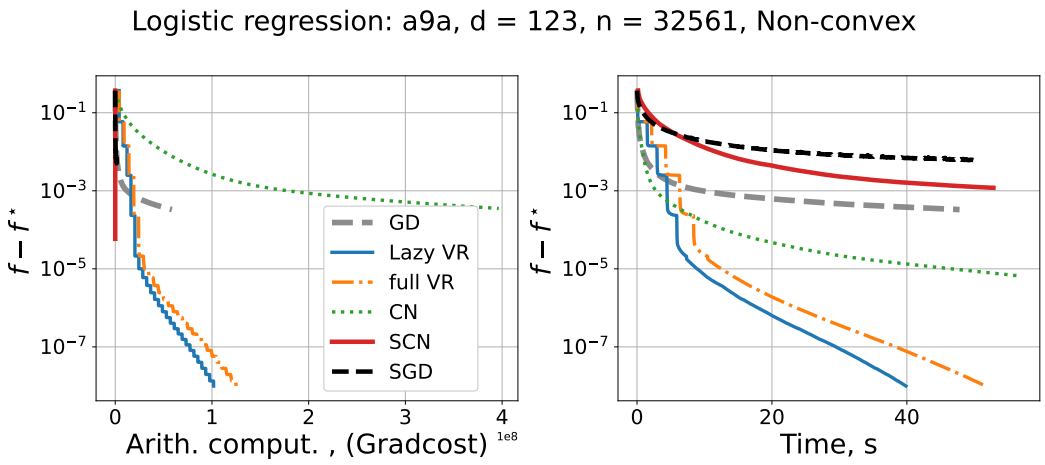

Figure 3: Comparison of the convergence of different algorithms. We see that using our approach we benefit a lot from the helper function h.

Second, we consider a simple diagonal neural network with L2 loss with data generated from a normal distribution. specifically, we want to minimize

$$f(\boldsymbol{x} := (\boldsymbol{u}, \boldsymbol{v})) = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{a}_i^\top \boldsymbol{u} \circ \boldsymbol{v} - b_i\|^2 + \frac{\lambda}{2}\|\boldsymbol{x}\|^2 \,,$$

where $\circ$ is the element-wise vector product.

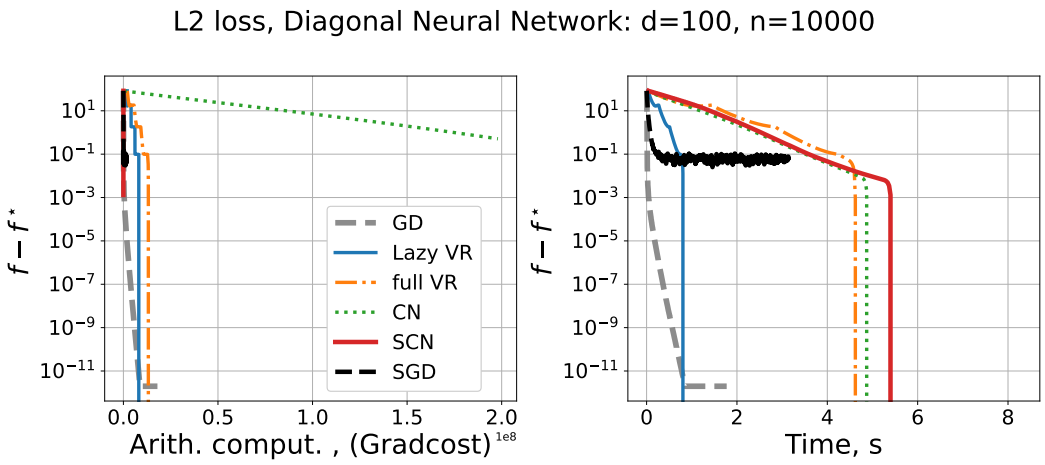

Figure 4: Comparison of the convergence of the different algorithms. Except for gradient descent ("GD") which performs very well in this case, again the same conclusions as in Figure 2 with respect to "Lazy VR" can be said.

Figure 4 shows again that compared to other second-order methods, "Lazy VR" has considerable time and computation savings. It also has a close performance to gradient descent with line search which performs very well in this case.

0.4 Reproducibility

We will make our code available with all the details necessary for reproducing our results in `https://anonymous.4open.science/r/Unified-Convergence-Theory-of-Cubic-Newton-s-method-E4C0/README.md`.

A Theoretical Preliminaries

We consider the general problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$$

Where f is twice differentiable with L-Lipschitz Hessian i.e.:

$$\|\nabla^2 f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{y})\| \leq L\|\boldsymbol{x}-\boldsymbol{y}\|, \qquad \forall \boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^d. \tag{13}$$

As a direct consequence of (13) (see [21, 20]) we have for all $\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^d$:

$$\|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})\| \leq \frac{L}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2, \tag{14}$$

$$|f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle\nabla f(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle - \frac{1}{2}\langle\nabla^2 f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle| \leq \frac{L}{6}\|\boldsymbol{y}-\boldsymbol{x}\|^3. \tag{15}$$

For $\boldsymbol{x}$ and $\boldsymbol{x}^+$ defined as in Equation (2) i.e.

$$\boldsymbol{x}^+ \quad \in \quad \arg\min_{\boldsymbol{y}\in\mathbb{R}^d}\Big\{ \Omega_{M,\boldsymbol{g},\boldsymbol{H}}(\boldsymbol{y},\boldsymbol{x}) := \langle\boldsymbol{g}, \boldsymbol{y}-\boldsymbol{x}\rangle + \tfrac{1}{2}\langle\boldsymbol{H}(\boldsymbol{y}-\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle + \tfrac{M}{6}\|\boldsymbol{y}-\boldsymbol{x}\|^3 \Big\}. \tag{16}$$

The optimality condition of (16) ensures that :

$$\langle\boldsymbol{g}, \boldsymbol{x}^+-\boldsymbol{x}\rangle + \langle\boldsymbol{H}(\boldsymbol{x}^+-\boldsymbol{x}), \boldsymbol{x}^+-\boldsymbol{x}\rangle + \frac{M}{2}r^3 = 0\,, \tag{17}$$

where we denoted $r = \|\boldsymbol{x}^+-\boldsymbol{x}\|$.

It is also known that the solution to (16) verifies :

$$\boldsymbol{H} + \frac{M}{2}r\mathbb{I} \succeq 0 \tag{18}$$

We start by proving the following Theorem

Theorem 5 *For any $\boldsymbol{x}\in\mathbb{R}^d$, let $\boldsymbol{x}^+$ be defined by* (2)*. Then, for $M\geq L$ we have:*

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^+) \quad \geq \quad \tfrac{1}{1008\sqrt{M}}\mu_M(\boldsymbol{x}^+) + \tfrac{M\|\boldsymbol{x}-\boldsymbol{x}^+\|^3}{72} - \tfrac{4\|\nabla f(\boldsymbol{x})-\boldsymbol{g}\|^{3/2}}{\sqrt{M}} - \tfrac{73\|\nabla^2 f(\boldsymbol{x})-\boldsymbol{H}\|^3}{M^2}.$$

Using (15) with $\boldsymbol{y}=\boldsymbol{x}^+$ and $\boldsymbol{x}=\boldsymbol{x}$ and for $M\geq L$ we have:

$$\begin{array}{rcl} f(\boldsymbol{x}^+) & \overset{(15)}{\leq} & f(\boldsymbol{x}) + \langle\nabla f(\boldsymbol{x}), \boldsymbol{x}^+-\boldsymbol{x}\rangle + \frac{1}{2}\langle\nabla^2 f(\boldsymbol{x})(\boldsymbol{x}^+-\boldsymbol{x}), \boldsymbol{x}^+-\boldsymbol{x}\rangle + \frac{L}{6}r^3 \\ & \overset{(17)+(18)}{\leq} & f(\boldsymbol{x}) - \frac{6M-4L}{24}r^3 + \langle\nabla f(\boldsymbol{x})-\boldsymbol{g}, \boldsymbol{x}^+-\boldsymbol{x}\rangle + \frac{1}{2}\langle(\nabla^2 f(\boldsymbol{x})-\boldsymbol{H})(\boldsymbol{x}^+-\boldsymbol{x}), \boldsymbol{x}^+-\boldsymbol{x}\rangle \\ & \overset{M\geq L}{\leq} & f(\boldsymbol{x}) - \frac{M}{12}r^3 + \langle\nabla f(\boldsymbol{x})-\boldsymbol{g}, \boldsymbol{x}^+-\boldsymbol{x}\rangle + \frac{1}{2}\langle(\nabla^2 f(\boldsymbol{x})-\boldsymbol{H})(\boldsymbol{x}^+-\boldsymbol{x}), \boldsymbol{x}^+-\boldsymbol{x}\rangle \end{array}$$

Using Young's inequality $xy \leq \frac{x^p}{p} + \frac{y^q}{q}\ \forall x,y\in\mathbb{R}\ \forall p,q>0$ s.t $\frac{1}{p}+\frac{1}{q}=1$ we have:

$$\langle\nabla f(\boldsymbol{x})-\boldsymbol{g}, \boldsymbol{x}^+-\boldsymbol{x}\rangle \leq \frac{M}{36}r^3 + \frac{2\sqrt{12}}{3\sqrt{M}}\|\nabla f(\boldsymbol{x})-\boldsymbol{g}\|^{3/2}\,,$$

and

$$\frac{1}{2}\langle(\nabla^2 f(\boldsymbol{x})-\boldsymbol{H})(\boldsymbol{x}^+-\boldsymbol{x}), \boldsymbol{x}^+-\boldsymbol{x}\rangle \leq \frac{M}{36}r^3 + \frac{72}{M^2}\|\nabla^2 f(\boldsymbol{x})-\boldsymbol{H}\|^3\,.$$

Mixing all these ingredients, we get

Lemma 2 *For any $M\geq L$, it holds*

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^+) \geq \frac{M}{36}r^3 - \frac{3}{\sqrt{M}}\|\nabla f(\boldsymbol{x})-\boldsymbol{g}\|^{3/2} - \frac{72}{M^2}\|\nabla^2 f(\boldsymbol{x})-\boldsymbol{H}\|^3. \tag{19}$$

Using (14) we have:

$$\|\nabla f(\boldsymbol{x}^+) - \boldsymbol{g} - \boldsymbol{H}(\boldsymbol{x}^+ - \boldsymbol{x}) + \boldsymbol{g} - \nabla f(\boldsymbol{x}) + (\boldsymbol{H} - \nabla^2 f(\boldsymbol{x}))(\boldsymbol{x}^+ - \boldsymbol{x})\| \leq \frac{L}{2} r^2,$$

applying the triangular inequality we get for $M \geq L$:

$$\begin{aligned}
\|\nabla f(\boldsymbol{x}^+)\| &\leq \frac{L}{2} r^2 + \|\boldsymbol{g} + \boldsymbol{H}(\boldsymbol{x}^+ - \boldsymbol{x})\| + \|\nabla f(\boldsymbol{x}) - \boldsymbol{g}\| + \|\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}\| r \\
&\leq \frac{L + 2M}{2} r^2 + \|\nabla f(\boldsymbol{x}) - \boldsymbol{g}\| + \frac{1}{2M} \|\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}\|^2 \\
&\leq \frac{3M}{2} r^2 + \|\nabla f(\boldsymbol{x}) - \boldsymbol{g}\| + \frac{1}{2M} \|\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}\|^2
\end{aligned}$$

By the convexity of $x \mapsto x^{3/2}$ we have for any $(a_i) \geq 0$: $(\sum_i a_i x_i)^{3/2} \leq (\sum_i a_i)^{1/2} \sum_i a_i x_i^{3/2}$, applying this to the above inequality we get

Lemma 3 *For any $M \geq L$, it holds*

$$\frac{1}{\sqrt{M}} \|\nabla f(\boldsymbol{x}^+)\|^{3/2} \leq 3Mr^3 + \frac{2}{\sqrt{M}} \|\nabla f(\boldsymbol{x}) - \boldsymbol{g}\|^{3/2} + \frac{1}{M^2} \|\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}\|^3 \tag{20}$$

We can also bound the smallest eigenvalue of the Hessian. Using the smoothness of the Hessian we have:

$$\begin{aligned}
\nabla^2 f(\boldsymbol{x}^+) &\succeq \nabla^2 f(\boldsymbol{x}) - L\|\boldsymbol{x}^+ - \boldsymbol{x}\| \mathbb{I} \\
&\succeq \boldsymbol{H} + \nabla^2 f(\boldsymbol{x}) - \boldsymbol{H} - Lr\mathbb{I} \\
&\succeq \boldsymbol{H} - \|\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}\| \mathbb{I} - Lr\mathbb{I} \\
&\overset{(18)}{\succeq} -\tfrac{Mr}{2}\mathbb{I} - \|\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}\| \mathbb{I} - Lr\mathbb{I}
\end{aligned}$$

Which means for $M \geq L$ we have:

$$-\lambda_{min}(\nabla^2 f(\boldsymbol{x}^+)) \leq \tfrac{3Mr}{2} + \|\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}\|$$

Then the convexity of $x \mapsto x^3$ leads to the following lemma :

Lemma 4 *For any $M \geq L$, it holds*

$$\frac{-\lambda_{min}(\nabla^2 f(\boldsymbol{x}^+))^3}{M^2} \leq 14Mr^3 + \frac{4}{M^2} \|\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}\|^3 \tag{21}$$

Now the quantity $\mu_M(\boldsymbol{x}) = \max(\|\nabla f(\boldsymbol{x})\|^{3/2}, \frac{-\lambda_{min}(\nabla^2 f(\boldsymbol{x}^+))^3}{M^{3/2}})$ which we can be bounded using Lemmas 3 and 4 :

$$\frac{1}{\sqrt{M}} \mu(\boldsymbol{x}^+) \leq 14Mr^3 + \frac{2}{\sqrt{M}} \|\nabla f(\boldsymbol{x}) - \boldsymbol{g}\|^{3/2} + \frac{4}{M^2} \|\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}\|^3. \tag{22}$$

Combining Lemma 2 and (22) we get the inequality given in Theorem 5:

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^+) \geq \frac{1}{1008\sqrt{M}} \mu_M(\boldsymbol{x}^+) + \frac{M}{72} r^3 - \frac{4}{\sqrt{M}} \|\nabla f(\boldsymbol{x}) - \boldsymbol{g}\|^{3/2} - \frac{73}{M^2} \|\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}\|^3.$$

B More on Section 3.1

B.1 Similarity using sampling

One common approach for constructing gradient and Hessian estimates is sub-sampling. The idea behind sub-sampling is simple: for an objective of the form in (1), we randomly sample two batches $\mathcal{B}_g$ and $\mathcal{B}_h$ of sizes b_g and b_h consecutively from the distribution $\mathcal{D}$ and define:

$$\boldsymbol{g}_{t,\mathcal{B}_g} = \frac{1}{b_g} \sum_{i \in \mathcal{B}_g} \nabla f_i(\boldsymbol{x}_t) \quad \text{and} \quad \boldsymbol{H}_{t,\mathcal{B}_h} = \frac{1}{b_h} \sum_{i \in \mathcal{B}_h} \nabla^2 f_i(\boldsymbol{x}_t) \tag{23}$$

In this particular scenario, the "elementary" estimates $\nabla f(\boldsymbol{x}_t, \zeta)$ and $\nabla^2 f(\boldsymbol{x}_t, \zeta)$ are unbiased, and we will assume that they satisfy $\mathbb{E}_i \|\nabla f(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x})\|^2 \leq \sigma_g^2$ and $\mathbb{E}_i \|\nabla^2 f(\boldsymbol{x}) - \nabla^2 f_i(\boldsymbol{x})\|^3 \leq \sigma_h^3$,.

Lemma 5 *For the estimators defined in* (23) *we have:*

$$\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \boldsymbol{g}_{t,\mathcal{B}_g}\|^2 \leq \frac{\sigma_g^2}{b_g} \quad \textit{and} \quad \mathbb{E}\|\nabla^2 f(\boldsymbol{x}_t) - \boldsymbol{H}_{t,\mathcal{B}_h}\|^3 \leq \mathcal{O}\big(\log(d)^{3/2}\frac{\sigma_h^3}{b_h^{3/2}}\big),$$

where $\mathcal{O}$ hides constant multiplicative factors.

Lemma 5 demonstrates how the utilization of batching can decrease the noise. To simplify things, we can keep in mind this straightforward rule:

If we employ a batch of size b_a, then we need to modify σ_a by $\frac{\sigma_a}{\sqrt{b_a}}$ for $a \in \{g, h\}$.

Lemma 5 is based on the following two Lemmas :

Lemma 6 ***(Lyapunov's inequality)*** *For any random variable X and any $0 < s < t$ we have*

$$\mathbb{E}[|X|^s]^{1/s} \leq \mathbb{E}[|X|^t]^{1/t}.$$

and

Lemma 7 ***Suppose that*** *$q \geq 2$, $p \geq 2$, and fix $r \geq \max(q, 2\log(p))$. Consider i.i.d. random self-adjoint matrices $Y_1, \cdots, Y_N$ with dimension $p \times p$, $\mathbb{E}[Y_i] = 0$. It holds that:*

$$\big[\mathbb{E}[\|\sum_{i=1}^N Y_i\|_2^q]\big]^{1/q} \leq 2\sqrt{er}\|\big(\sum_{i=1}^N \mathbb{E}[Y_i^2]\big)^{1/2}\|_2 + 4er\mathbb{E}[\max_i \|Y_i\|_2^q]^{1/q}.$$

Lemma 7 is taken from [36].

Now if we have $X_1, \cdots, X_b \in \mathbb{R}^d$, b i.i.d vector-valued random variables such that $\mathbb{E}[X_i] = \mu$ and $\mathbb{E}[\|X_i - \mu\|^2] \leq \sigma^2$ then by applying Lemma 6 we get :

$$\mathbb{E}[\|\frac{1}{b}\sum_i X_i - \mu\|^{3/2}] \leq \mathbb{E}[\|\frac{1}{b}\sum_i X_i - \mu\|^2]^{3/4} \leq \frac{\sigma^{3/2}}{b^{3/4}}.$$

When we have b i.i.d matrix-valued random variables $Y_1, \cdots, Y_b \in \mathbb{R}^{d\times d}$ such that $\mathbb{E}[Y_i] = \mu$, $\mathbb{E}[\|Y_i - \mu\|^2] \leq \sigma_2^2$ and $\mathbb{E}[\|Y_i - \mu\|^3] \leq \sigma_3^3$ (by Jensen's inequality $\sigma_2 \leq \sigma_3$), then by applying Lemma 7 we get:

$$\begin{aligned}\mathbb{E}[\|\tfrac{1}{b}\textstyle\sum_i Y_i - \mu\|^3] &\leq \left(2\sqrt{\tfrac{2e\log(d)}{b}}\sigma_2 + \tfrac{8e\log(d)}{b}\sigma_3\right)^3 \\ &= \mathcal{O}(\tfrac{\sigma_3^3}{b^{3/2}})\end{aligned}$$

These last two inequalities are identical to the statement of Lemma 5.

B.2 Proof of Theorem 1

We use here $\delta_1 = \sigma_g$ and $\delta_2 = \sigma_h$.

Combining both Theorem 5, Assumption 2 and Lemma 6 we get:

$$\begin{aligned}\mathbb{E}f(\boldsymbol{x}_t) - \mathbb{E}f(\boldsymbol{x}_{t+1}) &\geq \tfrac{1}{1008\sqrt{M}}\mathbb{E}\mu_M(\boldsymbol{x}_{t+1}) - \tfrac{4}{\sqrt{M}}\mathbb{E}\|\nabla f(\boldsymbol{x}_t) - \boldsymbol{g}_t\|^{3/2} - \tfrac{73}{M^2}\mathbb{E}\|\nabla^2 f(\boldsymbol{x}_t) - \boldsymbol{H}_t\|^3 \\ &\underset{Lemma\ 6}{\geq} \tfrac{1}{1008\sqrt{M}}\mathbb{E}\mu_M(\boldsymbol{x}_{t+1}) - \tfrac{4}{\sqrt{M}}\sigma_g^{3/2} - \tfrac{73}{M^2}\sigma_h^3\end{aligned}$$

By summing the above inequality from $t = 0$ to $t = T - 1$ and rearranging we get:

$$\frac{1}{1008T}\sum_{t=1}^T \mathbb{E}\mu_M(\boldsymbol{x}_t) \leq \sqrt{M}\frac{\mathbb{E}f(\boldsymbol{x}_0) - \mathbb{E}f(\boldsymbol{x}_T)}{T} + 4\sigma_g^{3/2} + \frac{73}{M^{3/2}}\sigma_h^3$$

All is left is to use the fact that $\mathbb{E}f(\boldsymbol{x}_0) - \mathbb{E}f(\boldsymbol{x}_T) \leq \mathbb{E}f(\boldsymbol{x}_0) - f^\star = F_0$, and by definition of $\boldsymbol{x}_{out}$: $\mathbb{E}\mu_M(\boldsymbol{x}_{out}) = \frac{1}{T}\sum_{t=1}^T \mathbb{E}\mu_M(\boldsymbol{x}_t)$, thus :

$$\frac{1}{1008}\mathbb{E}\mu_M(\boldsymbol{x}_{out}) \leq \frac{\sqrt{M}F_0}{T} + \frac{73}{M^{3/2}}\sigma_h^3 + 4\sigma_g^{3/2}$$

C Helper

C.1 Proof of Lemma 1

We have

$$f = \frac{1}{n}\sum_{i=1}^{n} f_i$$

and we suppose that all the f_i's have $L-$Lipschitz Hessians, so f too has an $L-$Lipschitz Hessian. Thus $f_i - f$ has $2L-$Lipschitz Hessian.

Applying (1) and 14 to $f_i - f$ we get

$$\|\mathcal{G}(f_i, \boldsymbol{x}, \tilde{\boldsymbol{x}}) - \nabla f(\boldsymbol{x})\| \leq L\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2$$

and

$$\|\mathcal{H}(f_i, \boldsymbol{x}, \tilde{\boldsymbol{x}}) - \nabla^2 f(\boldsymbol{x})\| \leq 2L\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|$$

We note also the if i is chosen at random then $\mathbb{E}_i\mathcal{G}(f_i, \boldsymbol{x}, \tilde{\boldsymbol{x}}) = \nabla f(\boldsymbol{x})$ and $\mathbb{E}_i\mathcal{H}(f_i, \boldsymbol{x}, \tilde{\boldsymbol{x}}) = \nabla^2 f(\boldsymbol{x})$.

By using the properties of variance we have

$$\mathbb{E}_{\mathcal{B}}\|\mathcal{G}(f_{\mathcal{B}}, \boldsymbol{x}, \tilde{\boldsymbol{x}}) - \nabla f(\boldsymbol{x})\|^2 \leq \frac{\mathbb{E}_i\|\mathcal{G}(f_i, \boldsymbol{x}, \tilde{\boldsymbol{x}}) - \nabla f(\boldsymbol{x})\|^2}{b} \leq \frac{L^2}{b}\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^4$$

Now all that is left is to apply Lemmas 6 and 7 we get

$$\mathbb{E}_{\mathcal{B}}\|\mathcal{G}(f_{\mathcal{B}}, \boldsymbol{x}, \tilde{\boldsymbol{x}}) - \nabla f(\boldsymbol{x})\|^3 \leq \frac{L^3}{b^{3/2}}\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^3$$

And

$$\begin{aligned}\mathbb{E}_{\mathcal{B}}\|\mathcal{H}(f_{\mathcal{B}}, \boldsymbol{x}, \tilde{\boldsymbol{x}}) - \nabla^2 f(\boldsymbol{x})\|^3 &\leq \left(2\sqrt{\frac{2e\log(d)}{b}} + \frac{8e\log(d)}{b}\right)^3 \mathbb{E}_i\|\mathcal{H}(f_i, \boldsymbol{x}, \tilde{\boldsymbol{x}}) - \nabla^2 f(\boldsymbol{x})\|^3 \\ &\leq \left(2\sqrt{\frac{2e\log(d)}{b}} + \frac{8e\log(d)}{b}\right)^3 L^3\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^3\end{aligned}$$

C.2 Proof of Theorem 2

We use Theorem 5 and denote denoting $r_{i+1} = \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|$ then by the definition of the similarity of h_1, h_2 to f we have:

$$\mathbb{E}f(\boldsymbol{x}_{sm+i}) - \mathbb{E}f(\boldsymbol{x}_{sm+i+1}) \geq \frac{1}{216\sqrt{M}}\mathbb{E}\mu_M(\boldsymbol{x}_{sm+i+1}) + \mathbb{E}\big[\frac{M}{72}r_{sm+i+1}^3 - (\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2})\|\boldsymbol{x}_{sm+i} - \boldsymbol{x}_{sm}\|^3\big]$$

We sum from $i = 0$ to $i = m - 1$

$$\mathbb{E}f(x_{sm}) - \mathbb{E}f(x_{(s+1)m}) \geq \sum_{i=0}^{m-1}\frac{1}{216\sqrt{M}}\mathbb{E}\mu_M(\boldsymbol{x}_{sm+i+1}) + \mathbb{E}\big[\frac{M}{72}r_{sm+i+1}^3 - (\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2})\|\boldsymbol{x}_{sm+i} - \boldsymbol{x}_{sm}\|^3\big]$$

We note that $\|\boldsymbol{x}_{sm+i} - \boldsymbol{x}_{sm}\| \leq \sum_{j=1}^{i-1} r_{sm+j}$, this means

$$\mathbb{E}f(x_{sm}) - \mathbb{E}f(x_{(s+1)m}) \geq \sum_{i=0}^{m-1}\frac{1}{216\sqrt{M}}\mathbb{E}\mu_M(\boldsymbol{x}_{sm+i+1}) + \mathbb{E}\big[\frac{M}{72}r_{sm+i+1}^3 - (\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2})(\sum_{j=1}^{i-1} r_{sm+j})^3\big]$$

We apply now the following inequality (from [10]) $\sum_{k=1}^{m-1}\Big(\sum_{i=1}^{k} r_i\Big)^3 \leq \frac{m^3}{3}\sum_{k=1}^{m-1} r_k^3$ true for positive numbers $\{r_k\}_{k\geq 1}$ and any $m \geq 1$. This inequality means :

$$\sum_{i=0}^{m-1}\big[\frac{M}{72}r_{sm+i+1}^3 - (\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2})(\sum_{j=1}^{i-1} r_{sm+j})^3\big] \geq (\frac{M}{72} - \frac{m^3}{3}(\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2}))\sum_{i=0}^{m-1} r_{sm+i+1}^3$$

The above quantity is thus positive if $\frac{M}{72} - \frac{m^3}{3}\left(\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2}\right) \geq 0$.

Equivalently, for M satisfying

$$\boxed{4(\tfrac{\delta_1}{M})^{3/2} + 73(\tfrac{\delta_2}{M})^3 \leq \tfrac{1}{24m^3}} \tag{24}$$

We have:

$$\mathbb{E}f(x_{sm}) - \mathbb{E}f(x_{(s+1)m}) \geq \tfrac{m}{216\sqrt{M}} \tfrac{1}{m} \textstyle\sum_{i=0}^{m-1} \mathbb{E}\mu(x_{sm+i+1}).$$

We sum from $s = 0$ to $s = S - 1$ which gives :

$$\frac{1}{216Sm} \sum_{s=0}^{S-1} \sum_{i=0}^{m-1} \mathbb{E}\mu(x_{sm+i+1}) \leq \frac{\sqrt{M}(f(x_0) - f^\star)}{Sm}$$

And thus by definition of $\boldsymbol{x}_{out}$ we have:

$$\mathbb{E}\mu(x_{out}) \leq 216 \frac{\sqrt{M}(f(x_0) - f^\star)}{Sm}$$

D Gradient dominated functions

D.1 Examples of gradient-dominated functions

Let us provide several main examples of functions satisfying (11):

Example 1 *Let f be convex on a bounded convex set Q of diameter D, and let solution $\boldsymbol{x}^\star$ to* (1) *belong to Q. Then, we have:*

$$f(\boldsymbol{x}) - f^\star \quad \leq \quad \langle \nabla f(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{x}^\star \rangle \quad \leq \quad D\|\nabla f(\boldsymbol{x})\|, \qquad \forall \boldsymbol{x} \in Q.$$

Therefore, f is $(D, 1)$-gradient dominated.

Example 2 *Let f be uniformly convex of degree $s \geq 2$ with some constant $\sigma > 0$:*

$$f(\boldsymbol{y}) \quad \geq \quad f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \tfrac{\sigma}{s}\|\boldsymbol{y} - \boldsymbol{x}\|^s, \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d.$$

Then, f is $\left(\frac{s-1}{s}(\frac{1}{\sigma})^{\frac{1}{s-1}}, \frac{s}{s-1}\right)$-gradient dominated (see, e.g. [11]).

In particular, uniformly convex functions of degree $s = 2$ are known as *strongly convex*, and we see that they satisfy condition (11) with $\tau = \frac{1}{2\sigma}$ and $\alpha = 2$. However, the function class (11) is much wider and it includes also some problems with *non-convex objectives* ([21]).

D.2 Special cases of Theorem 3

For **Convex functions.** Theorem 3 implies that for $M = \max\left(L, \frac{\sigma_h T}{2D}\right)$ we have the rate

$$\mathbb{E}[f(\boldsymbol{x}_{out})] - f^\star = \mathcal{O}\left(\frac{LD^3}{T^2} + \frac{\sigma_h D^2}{T} + \sigma_g D\right). \tag{25}$$

Equation (25) has been obtained by [1] but under the much stronger assumption of almost surely bounded noise. Using the gradient and Hessian estimates in (23), for $\varepsilon > 0$ and $M = L$, to reach an ε-global minimum, we need at most $T = \mathcal{O}(\sqrt{\frac{LD^3}{\varepsilon}})$, $b_h = \mathcal{O}(\frac{\sigma_h^2 D}{L\varepsilon})$ and $b_g = \mathcal{O}(\frac{\sigma_g^2 D^2}{\varepsilon^2})$. In other words, we need at most $\mathcal{O}(\frac{\sigma_g^2 L^{1/2} D^{3/2}}{\varepsilon^{5/2}} + d\frac{\sigma_h^2 D^{5/2}}{L^{1/2}\varepsilon^{3/2}})$ *GradCost*.

s-uniformly convex functions. For this class of functions, using the estimates in (23) and for $\varepsilon > 0$ and $M = L$, we reach an ε-global minimum in at most $T = \mathcal{O}(\frac{\sqrt{L}}{s^2}\log(\frac{F_0}{\varepsilon}))$, $b_h = \mathcal{O}(\frac{\sigma_h^2}{s^{4/3}L\varepsilon^{2/3}})$ and $b_g = \mathcal{O}(\frac{\sigma_g^2}{s^{8/3}\varepsilon^{4/3}})$ or equivalently $\tilde{\mathcal{O}}(\frac{\sigma_g^2\sqrt{L}}{s^{14/3}\varepsilon^{4/3}} + d\frac{\sigma_h^2}{s^{10/3}\sqrt{L}\varepsilon^{2/3}})$ *GradCost*.

μ-strongly convex functions. For this class of functions, for $M = L$, for any $\varepsilon > 0$ to get $\mathbb{E}[f(\boldsymbol{x}_{out})] - f^\star < \varepsilon$ we need at most $T = \mathcal{O}(t_0 + \log\log(\frac{\mu^3}{L^2\varepsilon}))$, $b_h = \mathcal{O}(\frac{\sigma_h^2}{L\sqrt{\mu\varepsilon}})$ and $b_g = \mathcal{O}(\frac{\sigma_g^2}{\mu\varepsilon})$ or $\tilde{\mathcal{O}}(\frac{t_0\sigma_g^2}{\mu\varepsilon} + d\frac{t_0\sigma_h^2}{L\sqrt{\mu\varepsilon}})$ *GradCost*.

D.3 A special case of Theorem 4

Since we have here many special cases depending on the value of α and the choices of the helpers, we will only consider the case of **convex functions** i.e. $\alpha = 1, \tau = D$, but it is easy to apply Theorem 4 to other cases (like uniformly convex and strongly convex functions).

Theorem 4 implies for convex functions the following:

$$\mathbb{E}[f(\boldsymbol{x}_{out})] - f^\star = \mathcal{O}\left(\frac{\delta_1 D^3}{S^2} + \frac{\delta_2 D^3}{S^2 m} + \frac{L D^3}{S^2 m^2}\right)$$

We have the following special cases based on the choice of the helper functions:

- Cubic Newton [21] corresponds to $\delta_1 = \delta_2 = 0$, we get indeed its known rate in the convex case. Under the **SOGEO** oracle, we reach an ε-global minimum in at most $\mathcal{O}(\frac{nd}{\sqrt{\varepsilon}})$ *GradCost*.
- Convex Lazy Cubic Newton (which was not considered in [10]) corresponds to $\delta_1 = 0, \delta_2 = L$, which gives $\mathbb{E}[f(\boldsymbol{x}_{out})] - f^\star = \mathcal{O}\left(\frac{LD^3}{S^2 m}\right)$. Under the **SOGEO** oracle, by choosing $m = d$ we reach an ε-global minimum in at most $\mathcal{O}(\frac{n\sqrt{d}}{\sqrt{\varepsilon}})$ *GradCost*.
- Convex variance reduced cubic Newton (also not considered by [36, 30]), corresponds to each time sampling $\mathcal{B}_g, \mathcal{B}_h$ of sizes b_g, b_h consecutively at random and setting $h_1 = \frac{1}{b_g}\sum_{i\in\mathcal{B}_g} f_i, h_2 = \frac{1}{b_h}\sum_{i\in\mathcal{B}_g} f_i$. According to Lemma 1 we have $\delta_1 = \frac{L}{\sqrt{b_g}}$ and $\delta_2 = \tilde{\mathcal{O}}(\frac{L}{\sqrt{b_h}})$ so for $b_g \sim m^4, b_h \sim m^2$ and $M = L$ we get $\mathbb{E}[f(\boldsymbol{x}_{out})] - f^\star = \mathcal{O}\left(\frac{LD^3}{S^2 m^2}\right)$. Again, Under the **SOGEO** oracle, by choosing $m = (nd)^{1/5}1_{d\le n^{2/3}} + n^{1/3}1_{d\ge n^{2/3}}$, we reach an ε-global-minimum in at most $\mathcal{O}\left(\frac{\min\left((nd)^{4/5}, n^{2/3}d+n\right)}{\sqrt{\varepsilon}}\right)$ *GradCost*.
- Variance reduced cubic newton with Lazy Hessians, in this case, by using sampling, we have $\delta_1 = \frac{L}{\sqrt{b_g}}$ and $\delta_2 = L$. If we take $m = (nd)^{1/3}1_{d\le\sqrt{n}} + d1_{d\ge\sqrt{n}}$ then we reach an ε-global-minimum in at most $\mathcal{O}\left(\frac{\min\left((nd)^{5/6}, n\sqrt{d}\right)}{\sqrt{\varepsilon}}\right)$ *GradCost*. Again this improves the complexity of Lazy Cubic Newton.

D.4 Proof of Theorem 3

From the previous proof we have

$$\mathbb{E}f(\boldsymbol{x}_t) - \mathbb{E}f(\boldsymbol{x}_{t+1}) \ge \frac{1}{1008\sqrt{M}}\mathbb{E}\|\nabla f(\boldsymbol{x}_{t+1})\|^{3/2} - \frac{4}{\sqrt{M}}\sigma_g^{3/2} - \frac{73}{M^2}\sigma_h^3$$

By the definition of (τ, α)-gradient dominated functions we have

$$f(\boldsymbol{x}_t) - f^\star \le \tau\|\nabla f(\boldsymbol{x}_t)\|^\alpha$$

So

$$\mathbb{E}\|\nabla f(\boldsymbol{x}_{t+1})\|^{3/2} \ge \mathbb{E}\big(\frac{f(\boldsymbol{x}_{t+1}) - f^\star}{\tau}\big)^{\frac{3}{2\alpha}}.$$

If $\alpha \le 3/2$, then by Jensen's inequality we have

$$\mathbb{E}\|\nabla f(\boldsymbol{x}_{t+1})\|^{3/2} \ge \big(\frac{\mathbb{E}f(\boldsymbol{x}_{t+1}) - f^\star}{\tau}\big)^{\frac{3}{2\alpha}}.$$

For $\alpha > 3/2$ we need to assume that $\mathbb{E}f(\boldsymbol{x}_t) - f^\star \le \tau\mathbb{E}[\|\nabla f(\boldsymbol{x}_t)\|]^\alpha$ which will give us also $\mathbb{E}\|\nabla f(\boldsymbol{x}_{t+1})\|^{3/2} \ge \big(\frac{\mathbb{E}f(\boldsymbol{x}_{t+1}) - f^\star}{\tau}\big)^{\frac{3}{2\alpha}}$.

We will consider the sequence $F_t = \mathbb{E}f(\boldsymbol{x}_t) - f^\star$ and denote $\gamma = \frac{3}{2\alpha}$, $C = \frac{1}{1008\sqrt{M}\tau^\gamma}$ and $a = \frac{4}{\sqrt{M}}\sigma_g^{3/2} + \frac{25}{M^2}\sigma_h^3$. Then the sequence (F_t) satisfies :

$$F_t - F_{t+1} \geq CF_{t+1}^{\gamma} - a \tag{26}$$

- Case $\gamma = 1$ then

$$F_{t+1} \geq \frac{F_t + a}{C+1}$$

So by recurrence, we have:

$$F_t \leq (1+C)^{-t}F_0 + \sum_{i=0}^{t-1}(1+C)^{-i}\frac{a}{1+C} \leq (1+C)^{-t}F_0 + \frac{a}{C}$$

We note that $(1+C)^{-t} \leq \exp(\frac{-Ct}{1+C})$ So

$$F_t \leq \exp(\frac{-Ct}{1+C})F_0 + \frac{a}{C}$$

- Case $\gamma \in (1,2]$ then let $\tilde{F}_t = \frac{F_t}{C^{1/(1-\gamma)}}$ and $\tilde{a} = \frac{a}{C^{1/(1-\gamma)}}$ for which we have

$$\tilde{F}_t - \tilde{F}_{t+1} \geq \tilde{F}_{t+1}^{\gamma} - \tilde{a}$$

Now let $x = \tilde{a}^{1/\gamma}$, and $\delta_t = \tilde{F}_t - x$ so :

$$\delta_t - \delta_{t+1} \geq (\delta_{t+1} + x)^{\gamma} - x^{\gamma} \geq \delta_{t+1}^{\gamma}$$

Where we used in the last inequality the fact that $(x+y)^{\gamma} \geq x^{\gamma} + y^{\gamma}$ for $\gamma \geq 1$ and $x, y \geq 0$.

All in all $\delta_t = \frac{F_t - (\frac{a}{C})^{1/\gamma}}{C^{1/(1-\gamma)}}$ and

$$\delta_t - \delta_{t+1} \geq \delta_{t+1}^{\gamma}$$

If $\delta_{t+1} \geq \delta_t/2$ then

$$\frac{1}{(\gamma-1)\delta_{t+1}^{\gamma-1}} - \frac{1}{(\gamma-1)\delta_t^{\gamma-1}} \geq \frac{\delta_t^{\gamma-1} - \delta_{t+1}^{\gamma-1}}{(\gamma-1)\delta_t^{\gamma-1}\delta_{t+1}^{\gamma-1}}$$

By concavity of $x \mapsto x^{\gamma-1}$ (since $\gamma \leq 2$) we get :

$$\frac{1}{(\gamma-1)\delta_{t+1}^{\gamma-1}} - \frac{1}{(\gamma-1)\delta_t^{\gamma-1}} \geq \frac{\delta_t - \delta_{t+1}}{(\gamma-1)\delta_t\delta_{t+1}^{\gamma-1}} \geq \frac{\delta_{t+1}}{\delta_t} \geq 1/2$$

If $\delta_{t+1} \leq \delta_t/2$ then

$$\frac{1}{(\gamma-1)\delta_{t+1}^{\gamma-1}} - \frac{1}{(\gamma-1)\delta_t^{\gamma-1}} \geq \frac{1}{(\gamma-1)\delta_t^{\gamma-1}}(2^{\gamma-1}-1) \geq \frac{2^{\gamma-1}-1}{(\gamma-1)\delta_0^{\gamma-1}}$$

Using the fact that (δ_t) is decreasing.

In all cases we have:

$$\frac{1}{(\gamma-1)\delta_{t+1}^{\gamma-1}} - \frac{1}{(\gamma-1)\delta_t^{\gamma-1}} \geq max(1/2, \frac{2^{\gamma-1}-1}{(\gamma-1)\delta_0^{\gamma-1}}) := D$$

By summing from $t = 0$ to $t = T-1$ we get :

$$\frac{1}{(\gamma-1)\delta_T^{\gamma-1}} \geq DT$$

In other words

$$\delta_T \leq \big(\frac{1}{(\gamma-1)DT}\big)^{\frac{1}{\gamma-1}}$$

- Case $\gamma < 1$: then we have

$$F_{t+1} \leq \big(\frac{F_t - F_{t+1} + a}{C}\big)^{1/\gamma}$$

By convexity of $x \mapsto x^{1/\gamma}$ we get

$$F_{t+1} \leq 2^{1/\gamma-1}\big(\frac{F_t - F_{t+1}}{C}\big)^{1/\gamma} + 2^{1/\gamma-1}\big(\frac{a}{C}\big)^{1/\gamma}$$

Let $\delta_t = \frac{F_t - 2^{1/\gamma-1}(\frac{a}{C})^{1/\gamma}}{2^{1/\gamma}C^{1/(1-\gamma)}}$ then we have

$$\delta_{t+1} \leq (\delta_t - \delta_{t+1})^{1/\gamma}$$

The sequence (δ_t) is decreasing thus $\delta_{t+1} \leq \delta_t^{1/\gamma}$ which guarantees a superlinear rate the moment $\delta_t < 1$.

In fact, we can show that at the beginning (δ_t) will decrease at least at a linear rate, and thus it will be at some point < 1.

We have $\frac{\delta_t}{\delta_{t+1}} \geq 1 + \frac{\delta_t - \delta_{t+1}}{\delta_{t+1}} \geq 1 + \frac{1}{\delta_{t+1}^{1-\gamma}} \geq 1 + \frac{1}{\delta_0^{1-\gamma}}$

Which means $\delta_{t+1} \leq (1 + \frac{1}{\delta_0^{1-\gamma}})^{-1}\delta_t = (1 - \frac{1}{1+\delta_0^{1-\gamma}})\delta_t \leq \exp(-\frac{1}{1+\delta_0^{1-\gamma}})\delta_t$.

To have $\delta_t \leq 1/2$ we need $t \geq t_0 = (1 + \delta_0^{1-\gamma})\log(2\delta_0)$ so that for $t \geq t_0$ we enjoy a superlinear rate and we have $\delta_t \leq \big(\frac{1}{2}\big)^{(\frac{1}{\gamma})^{t-t_0}}$

This finishes the proof.

D.5 Proof of Theorem 4

In Theorem 4 we made the choice of updating the snapshot in the following way $\tilde{\boldsymbol{x}}_{s+1} = \boldsymbol{x}_{\arg\min_{i\in\{0,\cdots,m-1\}} f(\boldsymbol{x}_{sm+i})}$ which means that $f(\tilde{\boldsymbol{x}}_{s+1}) \leq f(\boldsymbol{x}_{sm+i})$ for all $i \in \{0, \cdots, m-1\}$.

For $s \in \{0, \cdots, S-1\}$ and $i \in \{0, \cdots, m-1\}$ We have the following inequality

$$\mathbb{E}f(\boldsymbol{x}_{sm+i}) - \mathbb{E}f(\boldsymbol{x}_{sm+i+1}) \geq \frac{1}{216\sqrt{M}}\mathbb{E}\|\nabla f(\boldsymbol{x}_{sm+i+1})\|^{3/2} + \mathbb{E}\big[\frac{M}{72}r_{sm+i+1}^3 - \big(\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2}\big)\|\boldsymbol{x}_{sm+i} - \boldsymbol{x}_{sm}\|^3\big]$$

By definition of gradient-dominated functions we have $\mathbb{E}\|\nabla f(\boldsymbol{x}_{t+1})\|^{3/2} \geq \big(\frac{\mathbb{E}f(\boldsymbol{x}_{t+1}) - f^\star}{\tau}\big)^{\frac{3}{2\alpha}}$.

So

$$\begin{aligned}
\mathbb{E}f(\boldsymbol{x}_{sm+i}) - \mathbb{E}f(\boldsymbol{x}_{sm+i+1}) &\geq \frac{1}{216\sqrt{M}}\big(\frac{\mathbb{E}f(\boldsymbol{x}_{sm+i+1}) - f^\star}{\tau}\big)^{\frac{3}{2\alpha}} \\
&\quad + \mathbb{E}\big[\frac{M}{72}r_{sm+i+1}^3 - \big(\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2}\big)\|\boldsymbol{x}_{sm+i} - \boldsymbol{x}_{sm}\|^3\big] \\
&\geq \frac{1}{216\sqrt{M}}\big(\frac{\mathbb{E}f(\tilde{\boldsymbol{x}}_{s+1}) - f^\star}{\tau}\big)^{\frac{3}{2\alpha}} + \mathbb{E}\big[\frac{M}{72}r_{sm+i+1}^3 - \big(\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2}\big)\|\boldsymbol{x}_{sm+i} - \boldsymbol{x}_{sm}\|^3\big]
\end{aligned}$$

Summing the above inequality from $i = 0$ to $i = m-1$ and remarking that $\tilde{\boldsymbol{x}}_s = \boldsymbol{x}_{sm}$ we get

$$\mathbb{E}f(\tilde{\boldsymbol{x}}_s) - \mathbb{E}f(\boldsymbol{x}_{(s+1)m}) \geq \frac{m}{216\sqrt{M}}\big(\frac{\mathbb{E}f(\tilde{\boldsymbol{x}}_{s+1}) - f^\star}{\tau}\big)^{\frac{3}{2\alpha}} + \mathbb{E}\big[\sum_{i=0}^{m-1}\frac{M}{72}r_{sm+i+1}^3 - \big(\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2}\big)\|\boldsymbol{x}_{sm+i} - \boldsymbol{x}_{sm}\|^3\big]$$

By definition of $\tilde{\boldsymbol{x}}_{s+1}$ we have $f(\tilde{\boldsymbol{x}}_{s+1}) \leq f(\tilde{\boldsymbol{x}}_{sm+i})$ for all $i \in \{0, \ldots, m-1\}$ which leads to

$$\mathbb{E}f(\tilde{\boldsymbol{x}}_s) - \mathbb{E}f(\tilde{\boldsymbol{x}}_{s+1}) \geq \frac{m}{216\sqrt{M}}\big(\frac{\mathbb{E}f(\tilde{\boldsymbol{x}}_{s+1}) - f^\star}{\tau}\big)^{\frac{3}{2\alpha}} + \mathbb{E}\big[\sum_{i=0}^{m-1}\frac{M}{72}r_{sm+i+1}^3 - \big(\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2}\big)\|\boldsymbol{x}_{sm+i} - \boldsymbol{x}_{sm}\|^3\big]$$

For M satisfying (24) we have $\sum_{i=0}^{m-1} \frac{M}{72} r_{sm+i+1}^3 - \big(\frac{4\delta_1^{3/2}}{\sqrt{M}} + \frac{73\delta_2^3}{M^2}\big)\|\boldsymbol{x}_{sm+i} - \boldsymbol{x}_{sm}\|^3 \geq 0$ thus we have

$$\mathbb{E}f(\tilde{\boldsymbol{x}}_s) - \mathbb{E}f(\tilde{\boldsymbol{x}}_{s+1}) \geq \frac{m}{216\sqrt{M}}\big(\frac{\mathbb{E}f(\tilde{\boldsymbol{x}}_{s+1}) - f^\star}{\tau}\big)^{\frac{3}{2\alpha}}.$$

Let's define $F_s = \mathbb{E}f(\tilde{\boldsymbol{x}}_s) - f^\star$ then

$$F_s - F_{s+1} \geq C F_{s+1}^\gamma$$

which is a special case of the sequence 26 with $a = 0$, thus we can apply our findings from before and replace a by 0.