Journal of Machine Learning Research 23 (2022) 1-38 Submitted 8/20; Revised 3/22; Published 3/22

Accelerating Adaptive Cubic Regularization of Newton's Method via Random Sampling

Xi Chen XC13@STERN.NYU.EDU
Stern School of Business
New York University
New York, NY 10012, USA

Bo Jiang ISYEBOJIANG@GMAIL.COM
Research Institute for Interdisciplinary Sciences
School of Information Management and Engineering
Shanghai University of Finance and Economics
Shanghai 200433, China

Tianyi Lin DARREN_LIN@BERKELEY.EDU
Department of Electrial Engineering and Computer Science
University of California, Berkeley
Berkeley, CA 94720, USA

Shuzhong Zhang ZHANGS@UMN.EDU
Department of Industrial and Systems Engineering
University of Minnesota
Minneapolis, MN 55455, USA

Editor: Sathiya Keerthi

Abstract

In this paper, we consider an unconstrained optimization model where the objective is a sum of a large number of possibly nonconvex functions, though overall the objective is assumed to be smooth and convex. Our bid to solving such model uses the framework of cubic regularization of Newton's method. As well known, the crux in cubic regularization is its utilization of the Hessian information, which may be computationally expensive for large-scale problems. To tackle this, we resort to approximating the Hessian matrix via sub-sampling. In particular, we propose to compute an approximated Hessian matrix by either *uniformly* or *non-uniformly* sub-sampling the components of the objective. Based upon such sampling strategy, we develop accelerated adaptive cubic regularization approaches and provide theoretical guarantees on global iteration complexity of $\mathcal{O}(\epsilon^{-1/3})$ with high probability, which matches that of the original accelerated cubic regularization methods Jiang et al. (2020) using the *full* Hessian information. Interestingly, we also show that in the worst case scenario our algorithm still achieves an $\mathcal{O}(\epsilon^{-5/6}\log(\epsilon^{-1}))$ iteration complexity bound. The proof techniques are new to our knowledge and can be of independent interets. Experimental results on the regularized logistic regression problems demonstrate a clear effect of acceleration on several real data sets.

Keywords: Sum of nonconvex functions; acceleration; parameter-free adaptive algorithm; cubic regularization; Newton's method; random sampling; iteration complexity.

©2022 Xi Chen and Bo Jiang and Tianyi Lin and Shuzhong Zhang.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v23/20-910.html.

1. Introduction

In this paper, we consider the following *finite-sum* convex optimization problem:

$$f^* := \min_{\mathbf{x}\in\mathbb{R}^d} \; f(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}), \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is *smooth* and *convex*, while each component function $f_i : \mathbb{R}^d \to \mathbb{R}$ is *smooth* but possibly *nonconvex*. In addition, we assume $f^* > -\infty$. A variety of machine learning and statistics applications can be cast into problem (1) where f_i is interpreted as the loss of the i-th observation, e.g., Friedman et al. (2001); Sra et al. (2012); Kulis (2013); Bottou et al. (2018); Goodfellow et al. (2016). An important special case of problem (1) is

$$\min_{\mathbf{x}\in\mathbb{R}^d} \; f(\mathbf{x}) := \min_{\mathbf{x}\in\mathbb{R}^d} \; \left[\frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{a}_i^\top \mathbf{x})\right], \tag{2}$$

where $f_i : \mathbb{R} \to \mathbb{R}$ and $\mathbf{a}_i$ is the i-th observation. The formulation in Eq. (2) finds a wide range of applications. A typical example is the (regularized) maximum likelihood estimation for generalized linear models, which includes regularized least squares and regularized logistic regression. We refer the interested readers to Section 1.1 for more applications in form of Eq. (1) and Eq. (2).

Up till now, much of the efforts devoted to solving problem (1) has been on developing stochastic first-order approach (Shalev-Shwartz, 2016; Allen-Zhu and Yuan, 2016), due primarily to its simplicity nature in both theoretical analysis and practical implementation. However, stochastic gradient type algorithms are known to be sensitive to the conditioning of the problem and the parameters to be tuned in the algorithm (Xu et al., 2016). On the contrary, second-order optimization methods (Luenberger and Ye, 1984) have been shown to be generally robust (Roosta-Khorasani and Mahoney, 2019; Xu et al., 2016) and less sensitive to the parameter choices (Berahas et al., 2020; Xu et al., 2020a). A downside, however, is that the second-order type algorithms are more likely to prone to higher computational costs for large-scale problems, by nature of requiring the second-order information (viz. Hessian matrix). To alleviate this, one effective approach is the so-called sub-sampled second-order methods that approximate Hessian matrix via random sampling scheme (Drineas and Mahoney, 2018).

Recent trends in the optimization community tend to improve an existing method along two possible directions. The first direction of improvement is *acceleration*. Nesterov (1983, 2004) pioneered the study of accelerated gradient-based algorithms for convex optimization. For stochastic convex optimization, Lan (2012) developed an accelerated stochastic gradient-based algorithm. Since then, various accelerated stochastic first-order methods have been proposed (see, e.g., Shalev-Shwartz and Zhang (2013); Frostig et al. (2015); Ghadimi and Lan (2016); Allen-Zhu (2017); Jain et al. (2018); Allen-Zhu (2018)). Despite its popularity and simplicity, the stochastic first-order approach may perform poorly for ill-conditioned instances (Roosta-Khorasani and Mahoney, 2019) and can be sensitive to certain algorithmic parameters such as the choices of stepsizes (Berahas et al., 2020). In contrast, there are limited results (Song and Liu, 2019; Ghadimi et al., 2017; Ye et al., 2020) on accelerated stochastic second-order approaches. The second direction of improvement is to investigate

adaptive optimization algorithms without ensuring the problem parameters such as the first and the second order Lipschitz constants. In view of implementation, it is desirable to design algorithms that adaptively adjust these parameters since they are usually unknown *a priori*. A typical example is adaptive gradient method (e.g. AdaGrad (Duchi et al., 2011)), which is popular in the machine learning community.

However, such improvements – though highly desirable due to their relevance in machine learning – are largely lacking in the context of stochastic or sub-sampling second-order algorithms. When the objective function f is non-convex, sub-sampling adaptive cubic regularized Newton's methods (Kohler and Lucchi, 2017; Xu et al., 2020b) are capable of reaching a second-order critical point within an iteration bound of $\mathcal{O}(\epsilon^{-3/2})$. However, we are unaware of any existing accelerated sub-sampling second-order methods that are fully independent of problem parameters while maintaining superior convergence rate. Recall that Nesterov (2008) proposed an accelerated cubic regularized Newton's method with provable overall iteration complexity of $\mathcal{O}(\epsilon^{-1/3})$ for convex optimization.

Therefore, a natural question raises:

> *Can one develop an adaptive and accelerated sub-sampling cubic regularized method with an iteration complexity of $\mathcal{O}(\epsilon^{-1/3})$?*

In this paper, we provide an affirmative answer to the above question. In particular, by modifying the algorithm in our previous work Jiang et al. (2020), we manage to develop a novel sub-sampled cubic regularization method that is adaptive and accelerated. The advantages of the proposed approach inherited from that in Jiang et al. (2020) include: the algorithms are fully adaptive without requiring any problem parameters, and the cubic regularized sub-problem in the algorithms is allowed to be solved inexactly (see Condition 3.1) with some easy-to-satisfy approximation conditions similar to that in Birgin et al. (2017) and Jiang et al. (2020). In contrast with the algorithms in Jiang et al. (2020), we use the sub-sampled Hessian rather than the full Hessian in the cubic sub-problem to reduce the per-iteration computational cost, and the sub-sampled size gradually increases from a very small initial set, leading to a significant computational savings at the beginning steps of the algorithms. Moreover, we show that our proposed algorithm has the global convergence rate of $\mathcal{O}(\epsilon^{-1/3})$ with high probability (Theorem 9), which matches its deterministic counterparts (Jiang et al., 2020), requiring the availability of the *full* Hessian information. Although the issue of inexact Hessian has also been discussed in Jiang et al. (2020), the proposed Hessian approximation is based on the finite differences of the gradient, which is more expensive when n and d are large as shown in the numerical result section. In terms of the worst-case (i.e., when the error of the sub-sampled Hessian can not be controlled) performance of our algorithm, we show that it has a guarantee of $\mathcal{O}(\epsilon^{-5/6}\log(\epsilon^{-1}))$ iteration bound (Theorem 14). It is worth mentioning that the sub-sampled strategy is only adopted in approximating the Hessian matrix, while the true gradient is counted exactly. The merit of our method is particularly clear when both n and d are large (see Figure 2). Another advantage of counting the full gradient is that our algorithm has a worst-case performance guarantee (i.e., Theorem 14) in addition to the standard high probability result.

1.1 Examples

In this subsection, we provide a few examples in the form of Eq. (1) and Eq. (2) arising from applications of machine learning. Examples for convex component functions are well known, e.g. the regularized least squares problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left[\left(\mathbf{a}_i^\top \mathbf{x} - R(\mathbf{a}_i)\right)^2 + \lambda\|\mathbf{x}\|^2\right],$$

and the regularized logistic regression,

$$\min_{\mathbf{x}\in\mathbb{R}^d} \ f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left[\ln\left(1+\exp\left(-R(\mathbf{a}_i)\cdot\mathbf{a}_i^\top\mathbf{x}\right)\right) + \lambda\|\mathbf{x}\|^2\right],$$

where $\mathbf{a}_i \in \mathbb{R}^d$ and $R(\mathbf{a}_i)$ denote the feature and response of the i-th data point respectively. To be more specific, we have $R(\mathbf{a}_i) \in \mathbb{R}$ for the least squares loss, and $R(\mathbf{a}_i) \in \{-1, +1\}$ for logistic regression. The parameter $\lambda > 0$ is known as the regularization parameter.

Below we shall provide some examples where certain components in the finite sum may be nonconvex. Consider for instance the *nonconvex support vector machine* (Mason et al., 2000; Wang et al., 2017), where the objective function takes the form of

$$\min_{\mathbf{x}\in\mathbb{R}^d} \ f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n}\left[1-\tanh\left(R(\mathbf{a}_i)\cdot\mathbf{a}_i^\top\mathbf{x}\right) + \lambda\left\|\mathbf{x}\right\|^2\right],$$

which is an instance of (1) with

$$f_i(\mathbf{x}) = 1-\tanh\left(R(\mathbf{a}_i)\cdot\mathbf{a}_i^\top\mathbf{x}\right) + \lambda\left\|\mathbf{x}\right\|^2.$$

Indeed, for some choice of $\lambda > 0$, the objective is convex but a few component functions may be nonconvex.

Another example comes from *principal component analysis (PCA)*. Consider a set of n data vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$ in $\mathbb{R}^d$ and the normalized co-variance matrix $A = \frac{1}{n}\sum_{j=1}^{n}\mathbf{a}_j\mathbf{a}_j^\top$, PCA aims to find the leading principal component. Garber and Hazan (2015) proposed a new efficient optimization for PCA by reducing the problem to solving a small number of convex optimization problems, where a critical subroutine in the method is to solve

$$\min_{\mathbf{x}\in\mathbb{R}^d} \ \frac{1}{2}\mathbf{x}^\top(\mu\mathbb{I}-A)\,\mathbf{x} + b^\top\mathbf{x} = \min_{\mathbf{x}\in\mathbb{R}^d}\frac{1}{n}\sum_{j=1}^{n}\left[\frac{1}{2}\mathbf{x}^\top\left(\mu\mathbb{I}-\mathbf{a}_j\mathbf{a}_j^\top\right)\mathbf{x} + b^\top\mathbf{x}\right],$$

where μ is larger than or equal to the maximum eigenvalue of A. Although the above formulation is convex optimization, component functions in the above optimization problem may be nonconvex.

1.2 Related Works

The literature on the acceleration of second-order or higher-order methods for convex optimization is somewhat limited as compared to its first-order counterpart. Nesterov (2008)

improved the overall iteration complexity for convex optimization from $O(\epsilon^{-1/2})$ to $O(\epsilon^{-1/3})$ by means of the so-called cubic regularized Newton's method, and further accelerated it to $O(\epsilon^{-1/(p+1)})$ (Nesterov, 2019) by utilizing up to p-th order derivative information. Monteiro and Svaiter (2012) and Monteiro and Svaiter (2013) proposed the Newton proximal extragradient method (A-HPE) and its acceleration, which achieved an improved iteration complexity of $O(\epsilon^{-2/7})$. Recently, Arjevani et al. (2019) showed that $O(\epsilon^{-2/7})$ is actually a lower bound for the second-order methods to solve convex optimization, and thus A-HPE method is an optimal second-order method. Motivated by Monteiro and Svaiter's work, three groups of researchers independently proposed and analyzed some optimal high-order methods achieving the iteration complexity of $O(\epsilon^{-2/(3p+1)})$ (Gasnikov et al., 2019; Bubeck et al., 2019; Jiang et al., 2021). However, a bisection search procedure is necessary in each iteration of all these methods (Monteiro and Svaiter, 2013; Gasnikov et al., 2019; Bubeck et al., 2019; Jiang et al., 2021), and the total number of subproblems solved at each bisection step is bounded by a logarithmic factor in the given precision. On the other hand, the missing factor in the complexity estimate for the accelerated cubic regularized Newton's method is in the order of $O(\epsilon^{-1/21})$. As demonstrated by Nesterov (2019), the additional logarithmic factors in the complexity bound of A-HPE method will definitely overshadow its tiny superiority in the convergence rate. From the practical efficiency point of view, the acceleration second-order scheme presented in Nesterov (2008) and Monteiro and Svaiter (2013) are not easily implementable, since they assume the knowledge of some Lipschitz constant of the Hessian. To alleviate this, Jiang et al. (2020) incorporated an adaptive strategy (Cartis et al., 2011a,b) into Nesterov's approach (Nesterov, 2008, 2019), and further relaxed the criterion for solving each sub-problem while maintaining the same iteration complexity for convex optimization. However, the deterministic second-order method, e.g., the one proposed by Jiang et al. (2020), may be computationally costly as it requires the full second-order information.

The seminal work of Robbins and Monro (1951) triggered a burst of research interest on developing stochastic first-order methods. Regarding the second-order methods (in particular Newton's method), there has been a recent intensive research attention in designing their stochastic variants suitable for large-scale applications, e.g. stochastic quasi-Newton methods (Byrd et al., 2016; Schraudolph et al., 2007), stochastic cubic regularization method (Tripuraneni et al., 2018), randomized cubic regularization method (Doikov and Richtárik, 2018), stochastic trust region method (Blanchet et al., 2019), stochastic line search method (Paquette and Scheinberg, 2020), Hessian sketching (Pilanci and Wainwright, 2017; Cormode and Dickens, 2019) and sub-sampling methods (Agarwal et al., 2017; Byrd et al., 2011; Bollapragada et al., 2019; Erdogdu and Montanari, 2015; Kylasa et al., 2019; Liu et al., 2017; Yao et al., 2021; Li et al., 2020; Roosta-Khorasani and Mahoney, 2019; Xu et al., 2016). Note that all the works for finding the global minimizers on *sub-sampling methods* assume that all the component functions are convex. In terms of cubic regularized Newton's method for non-convex optimization, the adaptive regularization algorithms with inexact evaluation for both function and derivatives are considered in Bellavia et al. (2019). Kohler and Lucchi (2017) proposed a uniform sub-sampling strategy to approximate the Hessian matrix and the gradient, however, in each step of the algorithm the sample size for the approximation is unknown until the cubic subproblem in this iteration is solved. Xu et al. (2020b) resolved this issue by conducting appropriate uniform and

non-uniform sub-sampling strategies to construct Hessian approximations within the cubic regularization scheme and Yao et al. (2021) further proposed inexact variants of trust region and adaptive cubic regularization methods, which can be implemented in practice without any knowledge of unknowable problem-related quantities. The adaptive cubic regularization methods with dynamic inexact Hessian information for finite-sum minimization and stochastic optimization are studied in Bellavia et al. (2021) and Bellavia and Gurioli (2022) respectively. Zhang et al. (2021) managed to incoporate sub-sampling strategies into the variance reduction techniques. Under the framework of more general probabilistic models, some probabilistic convergence results for cubic regularization methods were established in Cartis and Scheinberg (2018). For convex optimization, Ghadimi et al. (2017) proposed an accelerated Newton's method with cubic regularization using inexact second-order information and such information could be obtained from a subsample strategy. However, their algorithm fails to retain the the iteration bound of $O(\epsilon^{-1/3})$, although the acceleration is indeed observed in the numerical experiments. Another recent work by Ye et al. (2020) resorted to Nesterov's acceleration to improve the convergence performance of second-order methods (approximate Newton), including regularized sub-sampled Newton, and provided nice empirical evaluation results. However, the acceleration is only achieved when the objective function is strongly convex. After the first version of this paper was published online, Song and Liu (2019) in the meanwhile studied an accelerated inexact proximal cubic regularized Newton's method that allows a composite objective: the sum of a smooth and a nonsmooth convex function. Their algorithm still assumes the knowledge of the Lipschitz constant, and has the iteration bound of $O(\epsilon^{-1/3})$ in the sense of expectation. It is worth noting that both Ghadimi et al. (2017) and Song and Liu (2019) assume the approximated Hessian is pre-given and satisfy certain nice properties that need be used in the analysis. In that regard, our algorithm allows a dynamic adjustment of the sample size of the approximated Hessian, which leads to a low per-iteration computational cost at certain stage of the algorithm. The resulting computational benefits are evidently observed (and some of which will be reported in this work) in the process of our numerical experiments.

1.3 Notations and Organization

Throughout the paper, we denote vectors by bold lower case letters, e.g., $\mathbf{x}$, and matrices by regular upper case letters, e.g., X. The transpose of a real vector $\mathbf{x}$ is denoted as $\mathbf{x}^\top$. For a vector $\mathbf{x}$, and a matrix X, $\|\mathbf{x}\|$ and $\|X\|$ denote the ℓ_2 norm and the matrix spectral norm, respectively. $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$ are respectively the gradient and the Hessian of f at $\mathbf{x}$, and $\mathbb{I}$ denotes the identity matrix. For two symmetric matrices A and B, $A \succeq B$ indicates that $A - B$ is symmetric positive semi-definite. The subscript, e.g., $\mathbf{x}_i$, denotes iteration counter. $\log(\alpha)$ denotes the natural logarithm of a positive number α. $\frac{0}{0} = 0$ is imposed for non-uniform setting. The inexact Hessian is denoted by $H(\mathbf{x})$, but for notational simplicity, we also use H_i to denote the inexact Hessian evaluated at the iterate $\mathbf{x}_i$ in iteration i, i.e., $H_i \triangleq H(\mathbf{x}_i)$. The calligraphic letter $\mathcal{S}$ denotes a collection of indices from $\{1, 2, \ldots, n\}$, with potential repeated items and its cardinality is denoted by $|\mathcal{S}|$.

The rest of the paper is organized as follows. In Section 2, we introduce the assumptions underlying this paper, and the tradeoff between the sample size and the accuracy of the resulting approximated Hessian. Then the sub-sampling accelerated cubic regularized

Newton's method is presented in Section 3. The probabilistic and worst case iteration complexity of the algorithm are analyzed in Section 4 and Section 5 respectively. In Section 6, we present some preliminary numerical results on solving regularized logistic regression, where the effect of acceleration together with low per-iteration computational cost are clearly observed. The details of most proofs can be found in the appendix.

2. Preliminaries

In this section, we first introduce the main definitions and assumptions used in the paper, and then present two lemmas on the construction of the inexact Hessian in random sampling.

2.1 Assumptions

Throughout this paper, we refer to the following definition of ϵ-optimality.

Definition 1 *(ϵ-optimality) Given $\epsilon \in (0,1)$, $\mathbf{x} \in \mathbb{R}^d$ is said to be an ϵ-optimal solution to problem (1), if*

$$f(\mathbf{x}) - f^\star \le \epsilon, \quad \text{or} \quad \|\nabla f(\mathbf{x})\|^2 \le \epsilon. \tag{3}$$

To proceed, we make the following standard assumption regarding the gradient and Hessian of the objective function f.

Assumption 2 *The objective function $f(\mathbf{x})$ in problem (1) is convex and twice differentiable. Each of $f_j(\mathbf{x})$ is possibly nonconvex but twice differentiable with the gradient and the Hessian being both Lipschitz continuous, i.e., there are $0 < L_j, \rho_j < \infty$ such that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ we have*

$$\|\nabla f_j(\mathbf{x}) - \nabla f_j(\mathbf{y})\| \le L_j \|\mathbf{x} - \mathbf{y}\|, \tag{4}$$

$$\|\nabla^2 f_j(\mathbf{x}) - \nabla^2 f_j(\mathbf{y})\| \le \rho_j \|\mathbf{x} - \mathbf{y}\|. \tag{5}$$

In the rest of the paper, we define $L = \max_j L_j > 0$ and $\bar{L} = \frac{1}{n}\sum_{j=1}^n L_j > 0$, and $\bar{\rho} = \frac{1}{n}\sum_{j=1}^n \rho_j$. A consequence of (4) is that

$$\|\nabla^2 f_j(\mathbf{x})\| \le L_j \ \text{ and } \ \|\nabla^2 f(\mathbf{x})\| \le \bar{L} \quad \forall\, \mathbf{x} \in \mathbb{R}^d. \tag{6}$$

We consider the following approximation of f evaluated at $\mathbf{x}_i$ with cubic regularization (Cartis et al., 2011a,b) in our algorithm:

$$m(\mathbf{s}; \mathbf{x}_i, \sigma_i) = f(\mathbf{x}_i) + \mathbf{s}^\top \nabla f(\mathbf{x}_i) + \frac{1}{2}\mathbf{s}^\top H(\mathbf{x}_i)\mathbf{s} + \frac{1}{3}\sigma_i \|\mathbf{s}\|^3, \tag{7}$$

where $\sigma_i > 0$ is a regularized parameter adjusted in the process as the algorithm progresses. Let $\mathbf{x}_0$ be the starting point of our algorithm and $\mathbf{x}^\star$ be an optimal solution of problem (1). Then sub-level set $\mathcal{L}(\mathbf{x}_0, \sigma_0) := \{\mathbf{x}_0 + \mathbf{s} \in \mathbb{R}^d \mid m(\mathbf{s}; \mathbf{x}_0, \sigma_0) \le m(0; \mathbf{x}_0, \sigma_0) = f(\mathbf{x}_0)\}$ at $\mathbf{x}_0$ with regularization parameter $\sigma = \sigma_0$ is bounded as the function $m(\mathbf{s}; \mathbf{x}_0, \sigma_0)$ is coercive. Hence, there is some $D \ge 1$ such that

$$\max_{\mathbf{x} \in \mathcal{L}(\mathbf{x}_0, \sigma_0)} \|\mathbf{x} - \mathbf{x}^\star\| \le D. \tag{8}$$

2.2 Random Sampling

When each f_i in (1) is convex, random sampling has been proven to be a very effective approach in reducing the computational cost; see Erdogdu and Montanari (2015); Roosta-Khorasani and Mahoney (2019); Bollapragada et al. (2019); Xu et al. (2016). In this subsection, we show that such random sampling can indeed be employed for the setting considered in this paper.

Suppose that the probability distribution of the sampling over the index set $\{1, 2, \ldots, n\}$ is $\mathbf{p} = \{p_i\}_{i=1}^{i=n}$ with $\text{Prob}(\xi = i) = p_i \geq 0$ for $i = 1, 2, \ldots, n$. Let $\mathcal{S}$ and $|\mathcal{S}|$ denote the sample collection and its cardinality respectively, and define

$$\tilde{H}(\mathbf{x}) = \frac{1}{n|\mathcal{S}|} \sum_{j \in \mathcal{S}} \frac{1}{p_j} \nabla^2 f_j(\mathbf{x}), \tag{9}$$

to be the sub-sampled Hessian. When n is very large, such random sampling can significantly reduce the per-iteration computational cost as $|\mathcal{S}| \ll n$. There are two sampling strategies in the literature: uniform sampling and non-uniform sampling Xu et al. (2020b). In the following, we review some technical result of each approach demonstrating how many samples are required to get an approximated Hessian within a given accuracy. The first one is to sample $\{1, 2, \ldots, n\}$ uniformly, i.e., $p_i = 1/n$. The lemma below is a simple restatement of Xu et al. (2020b, Lemma 16).

Lemma 3 *Suppose Assumption 2 holds for problem* (1)*. A uniform sampling with or without replacement is performed to form the sub-sampled Hessian. That is for* $\mathbf{x} \in \mathbb{R}^d$*, the matrix* $\tilde{H}(\mathbf{x})$ *is constructed from* (9) *with* $p_j = \frac{1}{n}$ *and sample size*

$$|\mathcal{S}| \geq \Theta^U(\hat{\epsilon}, \delta) := \frac{16L^2}{\hat{\epsilon}^2} \cdot \log\left(\frac{2d}{\delta}\right)$$

for given $0 < \hat{\epsilon}, \delta < 1$*, where* L *is defined as in Assumption 2. Then we have*

$$\text{Prob}(\|\tilde{H}(\mathbf{x}) - \nabla^2 f(\mathbf{x})\| \geq \hat{\epsilon}) < \delta.$$

In case problem (1) is endowed with more structures, then some more "informative" distribution may be constructed as opposed to simple uniform sampling. For instance, if it is in the form of (2), then we can introduce a bias in the probability distribution and pick those *relevant* f_i's carefully. As suggested in Xu et al. (2020b), we construct

$$p_j = \frac{|f_j''(\mathbf{a}_j^\top \mathbf{x})| \|\mathbf{a}_j\|^2}{\sum_{k=1}^n |f_j''(\mathbf{a}_k^\top \mathbf{x})| \|\mathbf{a}_k\|^2}, \tag{10}$$

where the absolute values are taken since f_j is possibly nonconvex. Next we restate Xu et al. (2020b, Lemma 17) below about the sampling complexity for the construction of approximated Hessian of problem (2).

Lemma 4 *Suppose Assumption 2 holds for problem* (2)*. A non-uniform sampling is performed to form the sub-sampled Hessian. That is for* $\mathbf{x} \in \mathbb{R}^d$*, the matrix* $\tilde{H}(\mathbf{x})$ *is constructed from* (9) *with* $\mathbf{p}$ *as defined in* (10) *and sample size*

$$|\mathcal{S}| \geq \Theta^N(\hat{\epsilon}, \delta) := \frac{4\bar{L}^2}{\hat{\epsilon}^2} \cdot \log\left(\frac{2d}{\delta}\right),$$

for given $0 < \hat{\epsilon}, \delta < 1$*, where* $\bar{L}$ *is defined in Assumption 2. Then, we have*

$$\text{Prob}(\|\tilde{H}(\mathbf{x}) - \nabla^2 f(\mathbf{x})\| \geq \hat{\epsilon}) < \delta.$$

Compared to Lemma 3, computing the sampling probability in Lemma 4 requires going through all data points, whose computational effort amounts to evaluating the full gradient once. However, the sampling complexity mainly comes from the sample size rather than the sampling probability. This is because the computational cost of forming the approximated Hessian matrix heavily depends on the sample size and such matrix is frequently sampled in our algorithm (i.e., sampled in every step of our algorithm). Moreover, the sample size provided by Lemma 4 could be smaller as $\bar{L} \leq L$. In this case, the non-uniform sampling is preferable where the distributions of L_j are skewed, i.e., some L_j are much larger than the others and $\bar{L} \ll L$. This advantage has been demonstrated by the practical performance of randomized coordinate descent method and sub-sampled Newton method (Qu and Richtárik, 2016a,b; Xu et al., 2016). Therefore, in this case, the computational savings stems from the smaller sample size dominates the cost of computing the sampling probability for the non-uniform sampling scheme.

Note that in the above two lemmas, the sample size is only proportional to the log of the failure probability, and thus we can use a very small failure per-iteration probability to guarantee the solution quality without increasing the sample size significantly. Although, the sample sizes in Lemma 3 and 4 is dependent on the Lipschitz constant, its exact value is not necessarily required and any of its upper bound would work. In addition, we provide worst-case analysis in Section 5, which guarantees the convergence of our algorithm regardless of the estimation quality of the Lipschitz constant.

3. Accelerated Adaptive Cubic Regularization of Newton's Method with Uniform and Nonuniform Sub-Sampling

3.1 The Algorithm

Now we propose the accelerated sub-sampling adaptive cubic regularization method as presented in Algorithm 1. In particular, we adopt a two-phase scheme, where the acceleration is implemented in Phase II. It is worth noting that a direct extension of the accelerated cubic regularization method under inexact Hessian information fails to maintain the theoretical convergence property (Ghadimi et al., 2017). Therefore, the two-phase scheme is necessary to establish the accelerated rate of convergence, where the first phase serves the purpose of finding a good starting point for acceleration. Phase I and Phase II are referred to as simple sub-sampling adaptive subroutine (SSAS) and accelerated sub-sampling adaptive subroutine (ASAS), respectively, and the details are described in Algorithm 2 and Algorithm 3. In particular, note that there are two counters of iterations in Algorithm 3. One is j that counts the generic iterations, and the other one is l for the successful iterations. In each generic iteration j of Algorithm 3, an approximate minimizer of the cubic model is computed. If the generic iteration is successful and early stopping is not activated, the auxiliar model is adaptively minimized in an inner loop for acceleration, and then the current approximation of the cubic model and the counter l for the successful iterations are updated. Otherwise, the current approximation is left unchanged and the coefficient σ of

Algorithm 1 Accelerated Subsampling Adaptive Cubic Regularized Newton's Method

Input: $\mathbf{x}_0 \in \mathbb{R}^d$, $\sigma_0 \geq \sigma_{\min} > 0$, $\tau_0 > 0$, $\gamma_2 > \gamma_1 > 1$, $\gamma_3 > 1$, $\eta > 0$, $\delta_0 \in (0,1)$, $\kappa_\theta \in (0,1)$, initial tolerance of Hessian approximation $\epsilon_0 = \min\{1, \frac{\|\nabla f(\mathbf{x}_0)\|}{3}\}$, and tolerance of the approximate solution ϵ.
Phase I (SSAS): $[\mathbf{x}_0^I, \sigma_0^I, \epsilon^I, T_1] = \mathsf{SSAS}(\mathbf{x}_0, \sigma_0, \epsilon_0, \epsilon, \gamma_1, \gamma_2, \delta_0, \kappa_\theta)$.
if $\|\nabla f(\mathbf{x}_0^I)\|^2 \leq \epsilon$ **then**
terminate Algorithm 1 [*early stop*], and return $\mathbf{x}_{out} = \mathbf{x}_0^{ASAS}$.
end if
Phase II (ASAS): $[\mathbf{x}_{out}, T_2, T_3] = \mathsf{ASAS}(\mathbf{x}_0^I, \sigma_0^I, \sigma_{\min}, \epsilon^I, \epsilon, \varsigma_0, \gamma_1, \gamma_2, \gamma_3, \eta, \delta_0, \kappa_\theta)$.
Let $T = T_1 + T_2 + T_3$ [*record the total iteration number*].
Output: an ε-optimal solution $\mathbf{x}_{out}$ and T.

the cubic regularization term is reduced. In the following, we elaborate on some key steps of these algorithms.

Constructing the cubic model: Given the iteration point $\mathbf{x}_i$, cubic regularized parameter σ_i, tolerance of Hessian approximation ϵ_i, the accuracy of the optimal solution ϵ, and overall failure probability δ_0. We adopt the notation $\mathsf{Cubic}(\mathbf{x}_i, \sigma_i, \epsilon_i, \epsilon, \delta_0)$ to denote the generator of the cubic model as follows:

$$\mathsf{Cubic}(\mathbf{x}_i, \sigma_i, \epsilon_i, \epsilon, \delta_0) \to f(\mathbf{x}_i) + \mathbf{s}^\top \nabla f(\mathbf{x}_i) + \frac{1}{2}\mathbf{s}^\top H(\mathbf{x}_i)\mathbf{s} + \frac{1}{3}\sigma_i \|\mathbf{s}\|^3, \tag{11}$$

where $H(\mathbf{x}_i) = \tilde{H}(\mathbf{x}_i) + \epsilon_i \mathbb{I}$, and $\tilde{H}(\mathbf{x}_i)$ is constructed according to (9) with sample size $|\mathcal{S}| \geq \Theta^U(\epsilon_i, \delta_0\epsilon^{1/3})$ for uniform sampling ($\Theta^N(\epsilon_i, \delta_0\epsilon^{1/3})$ for non-uniform sampling) such that

$$\|\nabla^2 f(\mathbf{x}_i) - \tilde{H}(\mathbf{x}_i)\| \leq \epsilon_i$$

with probability at least $1 - \delta_0\epsilon^{1/3}$. If the above inequality holds, the approximated Hessian $H(\mathbf{x}_i)$ in the cubic model is also a good estimation, i.e.,

$$\|\nabla^2 f(\mathbf{x}_i) - H(\mathbf{x}_i)\| \leq 2\epsilon_i. \tag{12}$$

In addition, the convexity of f implies that

$$H(\mathbf{x}_i) = \tilde{H}(\mathbf{x}_i) + \epsilon_i \mathbb{I} \succeq \nabla^2 f(\mathbf{x}_i) - \epsilon_i \mathbb{I} + \epsilon_i \mathbb{I} = \nabla^2 f(\mathbf{x}_i) \succeq 0. \tag{13}$$

with probability at least $1 - \delta_0\epsilon^{1/3}$.

Solving the cubic model: Recall that $m(\mathbf{s}; \mathbf{x}_i, \sigma_i)$ is the cubic σ_i-regularized function at $\mathbf{x}_i$ defined in (7). In each iteration, we approximately solve

$$\mathbf{s}_i \approx \operatorname*{argmin}_{\mathbf{s}\in\mathbb{R}^d} \; m(\mathbf{s}; \mathbf{x}_i, \sigma_i), \tag{14}$$

where $m(\mathbf{s}; \mathbf{x}_i, \sigma_i)$ is defined in (7) and the symbol "$\approx$" is quantified as follows:

Condition 3.1 *We call $\mathbf{s}_i$ to be an approximate solution of the subproblem – denoted as $\mathbf{s}_i \approx \operatorname{argmin}_{\mathbf{s}\in\mathbb{R}^d} \; m(\mathbf{s}; \mathbf{x}_i, \sigma_i)$ – for $\min_{\mathbf{s}\in\mathbb{R}^d} \; m(\mathbf{s}; \mathbf{x}_i, \sigma_i)$, if $m(\mathbf{s}_i; \mathbf{x}_i, \sigma_i) \leq m(0; \mathbf{x}_i, \sigma_i) = f(\mathbf{x}_i)$ and*

$$\|\nabla f(\mathbf{x}_i) + H(\mathbf{x}_i)\mathbf{s}_i + \sigma_i \|\mathbf{s}_i\| \mathbf{s}_i\| \leq \kappa_\theta \min\{\|\mathbf{s}_i\|^2, \|\nabla f(\mathbf{x}_i)\|\}, \tag{15}$$

where $0 < \kappa_\theta < 1$ is a pre-specified constant.

Algorithm 2 $\mathsf{SSAS}(\mathbf{x}_0, \sigma_0, \epsilon_0, \epsilon, \gamma_1, \gamma_2, \delta_0, \kappa_\theta)$

Initialization: Let the total iteration count $i = 0$.
Generate cubic model $m(\mathbf{s}; \mathbf{x}_0, \sigma_0)$ with $\mathsf{Cubic}(\mathbf{x}_0, \sigma_0, \epsilon_0, \epsilon, \delta_0)$ according to (11).
Let $\theta_0 = -1$.
while $\theta_i \leq 0$ **do**
  Compute $\mathbf{s}_i \in \mathbb{R}^d$ such that $\mathbf{s}_i \approx \mathrm{argmin}_{\mathbf{s}\in\mathbb{R}^d}\ m(\mathbf{s}; \mathbf{x}_0, \sigma_0)$ according to Condition 3.1;
  Compute $\theta_i = m(\mathbf{s}_i; \mathbf{x}_i, \sigma_i) - f(\mathbf{x}_i + \mathbf{s}_i)$.
  if $\theta_i > 0$ [*successful iteration*] **then**
    Let $\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{s}_i$, $\sigma_{i+1} = \sigma_i$,

$$\epsilon_{i+1} = \min\left\{\frac{\|\nabla f(\mathbf{x}_{i+1})\|}{6},\ \epsilon_0\right\} [\textit{update tolerance of Hessian approximation}].$$

    Update $i = i + 1$.
  else
    $\mathbf{x}_{i+1} = \mathbf{x}_i$, $\sigma_{i+1} \in [\gamma_1\sigma_i, \gamma_2\sigma_i]$, $\epsilon_{i+1} = \epsilon_i$, update $i = i + 1$.
  end if
end while
Let $T_1 = i$ [*record the iteration number*].
Return $\mathbf{x}_i, \sigma_i, \epsilon_i$.

Solving the auxiliary model: The acceleration in Phase II is achieved by minimizing an auxiliary model:

$$\psi_l(\mathbf{z}) = \psi_{l-1}(\mathbf{z}) + \frac{l(l+1)}{2}\left(f(\bar{\mathbf{x}}_{l-1}) + (\mathbf{z} - \bar{\mathbf{x}}_{l-1})^\top \nabla f(\bar{\mathbf{x}}_{l-1})\right) + \frac{1}{6}(\varsigma_l - \varsigma_{l-1})\|\mathbf{z} - \bar{\mathbf{x}}_0\|^3,$$

with $\psi_0(\mathbf{z}) = f(\mathbf{x}_0) + \frac{1}{6}\varsigma_0\|\mathbf{z} - \bar{\mathbf{x}}_0\|^3$. To be specific, $\psi_l(\mathbf{z})$ is used as a bridge to establish the iteration bounds in Theorem 8 and Theorem 13. Moreover, the minimizer of auxiliary model $\psi_l(\mathbf{z})$ has a closed-form expression (see Nesterov (2008) and Jiang et al. (2020)): $\bar{\mathbf{x}}_0 - \sqrt{\frac{2}{\varsigma_l\|\nabla\ell_l(\mathbf{z})\|}}\nabla\ell_l(\mathbf{z})$ with

$$\ell_l(\mathbf{z}) = \ell_{l-1}(\mathbf{z}) + \frac{l(l+1)}{2}\left(f(\bar{\mathbf{x}}_{l-1}) + (\mathbf{z} - \bar{\mathbf{x}}_{l-1})^\top \nabla f(\bar{\mathbf{x}}_{l-1})\right) \quad \text{and} \quad \ell_0(\mathbf{z}) = f(\mathbf{x}_0).$$

3.2 Overview of the Analysis

Recall in our algorithms that T_1 is the total number of iterations in Phase I, T_2 is the total number of solving the cubic model in Phase II, and T_3 is the total count of updating the parameter ς_l in the auxiliary model. Then the iteration complexity is established if we are able to bound T_1, T_2 and T_3. Before presenting the technical analysis, we sketch some major steps as follows,

1. Upper bound T_1 in Lemma 5 (Lemma 10 for worst case analysis) .

Algorithm 3 $\mathsf{ASAS}(\mathbf{x}_0, \sigma_0, \sigma_{\min}, \epsilon_0, \epsilon, \varsigma_0, \gamma_1, \gamma_2, \gamma_3, \eta, \delta, \kappa_\theta)$

Initialization: Let the total iteration count $i = 0$, the successful iteration count $l = 0$, the iteration count $k = 0$ of updating ς_l, and $\bar{\mathbf{x}}_0 = \mathbf{x}_0$.
Construct $\psi_0(\mathbf{z}) = f(\bar{\mathbf{x}}_0) + \frac{1}{6}\varsigma_0\|\mathbf{z} - \bar{\mathbf{x}}_0\|^3$, and compute $\mathbf{z}_0 = \operatorname{argmin}_{\mathbf{z}\in\mathbb{R}^d} \psi_0(\mathbf{z})$.
Let $\mathbf{y}_0 = \frac{1}{4}\bar{\mathbf{x}}_0 + \frac{3}{4}\mathbf{z}_0$ [*generate base point for the cubic model*].
Generate cubic model $m(\mathbf{s}; \mathbf{y}_0, \sigma_0)$ with $\mathsf{Cubic}(\mathbf{y}_0, \sigma_0, \epsilon_0, \epsilon, \delta_0)$ according to (11).
for $j = 0, 1, 2, \ldots,$ until convergence **do**
  Compute $\mathbf{s}_j \approx \operatorname{argmin}_{\mathbf{s}\in\mathbb{R}^d} m(\mathbf{s}; \mathbf{y}_l, \sigma_j)$ using Condition 3.1, and $\rho_j = -\frac{\mathbf{s}_j^\top \nabla f(\mathbf{y}_l + \mathbf{s}_j)}{\|\mathbf{s}_j\|^3}$.
  if $\rho_j \geq \eta$ [*successful iteration*] **then**
    $\bar{\mathbf{x}}_{l+1} = \mathbf{x}_{j+1} = \mathbf{y}_l + \mathbf{s}_j$, $\sigma_{j+1} \in [\sigma_{\min}, \sigma_j]$, and let

$$\epsilon_{j+1} = \min\left\{\frac{\|\nabla f(\mathbf{y}_l)\|}{4},\ \epsilon_0\right\}. \text{ [update tolerance of Hessian approximation]}$$

    if $\|\nabla f(\mathbf{x}_{j+1})\|^2 \leq \epsilon$ **then**
      terminate Algorithm 3 [*early stop*], and return $\mathbf{x}_{out} = \mathbf{x}_{j+1}$.
    end if
    Set $l = l + 1$, $\varsigma_l = \varsigma_{l-1}$, and compute $\mathbf{z}_l = \operatorname{argmin}_{\mathbf{z}\in\mathbb{R}^d}\ \psi_l(\mathbf{z})$.
    while $\psi_l(\mathbf{z}_l) < \frac{l(l+1)(l+2)}{6} f(\bar{\mathbf{x}}_l)$ **do**
      Set $\varsigma_l = \gamma_3\varsigma_l$, and $k = k + 1$ [*record the count of updating* ς_l].
      Update $\psi_l(\mathbf{z}) = \psi_{l-1}(\mathbf{z}) + \frac{l(l+1)}{2}[f(\bar{\mathbf{x}}_l) + (\mathbf{z} - \bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l)] + \frac{1}{6}(\varsigma_l - \varsigma_{l-1})\|\mathbf{z} - \bar{\mathbf{x}}_0\|^3$.
      Compute $\mathbf{z}_l = \operatorname{argmin}_{\mathbf{z}\in\mathbb{R}^d} \psi_l(\mathbf{z})$.
    end while
    Compute $\mathbf{y}_l = \frac{l}{l+3}\bar{\mathbf{x}}_l + \frac{3}{l+3}\mathbf{z}_l$ [*generate base point for the cubic model*].
    Generate cubic model $m(\mathbf{s}; \mathbf{y}_l, \sigma_{j+1})$ with $\mathsf{Cubic}(\mathbf{y}_l, \sigma_{j+1}, \epsilon_{j+1}, \epsilon, \delta_0)$ according to (11).
  else
    Let $\mathbf{x}_{j+1} = \mathbf{x}_j$, $\sigma_{j+1} \in [\gamma_1\sigma_j, \gamma_2\sigma_j]$;
  end if
end for
Let $T_2 = j + 1$ [*record the number of solving the cubic subproblem*].
Let $T_3 = k$ [*record the total number of updating* ς_l].
Return $\mathbf{x}_{out} = \bar{\mathbf{x}}_l$, T_2 and T_3.

2. Prove T_2 to be $|\mathcal{SC}|$ multiplied by some factors in Lemma 6 (Lemma 11 for worst case analysis), where $\mathcal{SC} = \{j \leq T_2 : j \text{ is a successful iteration}\}$ is the index set of all successful iterations in Phase II.

3. Upper bound T_3 in Lemma 7 (Lemma 12 for worst case analysis).

4. Upper bound $|\mathcal{SC}|$ in Theorem 8 (Theorem 13 for worst case analysis).

5. Put the pieces together, and prove the iteration bound in Theorem 9 (Theorem 14 for worst case analysis).

For probabilistic iteration complexity, the quantities in the bounds of T_1, T_2 and T_3 (except for $|\mathcal{SC}|$) depend only on the problem parameters (see Lemmas 5–7) thus do not affect the magnitude of the iteration bound. While for the worst-case iteration complexity, those quantities in Lemmas 10–12 depend on ϵ, i.e., the solution accuracy. Such dependence will eventually deteriorate the iteration complexity bound, such as the one presented in Theorem 14.

4. Probabilistic Iteration Complexity

Now we are in a position to provide iteration complexity analysis for Algorithm 1. We shall show that Algorithm 1 retains the iteration complexity of $O(\epsilon^{-1/3})$, the same as that of the non-adaptive version (Nesterov, 2008), even though the sub-problem is now only solved approximately with sub-sampled Hessian. In the following, to highlight the flow of our analysis, we shall present the contents of the key technical lemmas while relegating the proofs to the appendix.

We first provide Lemma 5 and Lemma 6, which describe the relation between the total iteration number in Algorithm 1 and the amount of successful iterations $|\mathcal{SC}|$ in Phase II.

Lemma 5 *Suppose in each iteration i of Algorithm 2, the sub-sampled Hessian $\tilde{H}(\mathbf{x}_i)$ satisfies*

$$\|\nabla^2 f(\mathbf{x}_i) - \tilde{H}(\mathbf{x}_i)\| \leq \epsilon_i. \tag{16}$$

Denoting $\bar{\sigma}_1^P = \max\{\sigma_0,\ 3\gamma_2 + 0.5\bar{\rho}\gamma_2,\ \gamma_2(L + \epsilon_0 + \kappa_\theta + \bar{\rho})\}$, it holds that

$$T_1 \leq \left\lceil 1 + \frac{1}{\log \gamma_1} \log\left(\frac{\bar{\sigma}_1^P}{\sigma_{\min}}\right) \right\rceil.$$

Lemma 6 *Suppose in each iteration j of Algorithm 3, the sub-sampled Hessian $\tilde{H}(\mathbf{x}_j)$ satisfies $\|\nabla^2 f(\mathbf{x}_j) - \tilde{H}(\mathbf{x}_j)\| \leq \epsilon_j$. Denoting*

$$\bar{\sigma}_2^P = \max\left\{\bar{\sigma}_1^P,\ \frac{\gamma_2\bar{\rho}}{2} + \gamma_2\kappa_\theta + \gamma_2\eta + 2\gamma_2,\ \gamma_2 L + \gamma_2\epsilon_0 + \gamma_2\bar{\rho} + 3\gamma_2\kappa_\theta + 2\gamma_2\eta\right\} > 0,$$

and $\mathcal{SC}$ to be the set of successful iterations in Algorithm 3, it holds that

$$T_2 \leq \left\lceil 1 + \frac{2}{\log \gamma_1} \log\left(\frac{\bar{\sigma}_2^P}{\sigma_{\min}}\right) \right\rceil |\mathcal{SC}|.$$

Then we estimate an upper bound on T_3: the total counts updating $\varsigma > 0$.

Lemma 7 *Suppose in each iteration j of Algorithm 3, the sub-sampled Hessian $\tilde{H}(\mathbf{x}_j)$ satisfies $\|\nabla^2 f(\mathbf{x}_j) - \tilde{H}(\mathbf{x}_j)\| \leq \epsilon_j$. It holds that*

$$\psi_l(\mathbf{z}_l) \geq \frac{l(l+1)(l+2)}{6} f(\bar{\mathbf{x}}_l) \tag{17}$$

when $\varsigma_l \geq \bar{\varsigma}^P := 8\eta^{-2}(\bar{\rho} + (2\kappa_\theta + 2)L + 2\bar{\sigma}_2^P + \kappa_\theta + 1)^3$, which further implies

$$T_3 \leq \left\lceil 1 + \frac{1}{\log(\gamma_3)} \log\left[\frac{8\left(\frac{\bar{\rho}}{2} + 2\kappa_\theta + L + 2\bar{\sigma}_2^P + 1\right)^3}{\eta^2 \varsigma_0}\right] \right\rceil.$$

In the rest of this section, the total number of iterations of the two subroutines (i.e. Algorithm 2 and Algorithm 3) is referred to as the iteration complexity of Algorithm 1. To continue our analysis, we prove the following theorem to provide a bound on the number of successful iterations in Algorithm 3.

Theorem 8 *Suppose in each iteration i of Algorithm 1, the sub-sampled Hessian $\tilde{H}(\mathbf{x}_i)$ satisfies* (16)*. Then the sequence $\{\bar{\mathbf{x}}_l,\ l = 0, 1, \ldots\}$ generated by Algorithm 3 satisfies*

$$\begin{aligned} & \frac{l(l+1)(l+2)}{6} f(\bar{\mathbf{x}}_l) \leq \psi_l(\mathbf{z}_l) \leq \psi_l(\mathbf{z}) \\ \leq\ & \frac{l(l+1)(l+2)}{6} f(\mathbf{z}) + 8\kappa_\theta D^3 + \frac{\bar{L}+\epsilon_0}{2}\|\mathbf{z}-\mathbf{x}_0\|^2 + \frac{\bar{\sigma}_1^P}{3}\|\mathbf{z}-\mathbf{x}_0\|^3 + \frac{\mathcal{G}l}{6}\|\mathbf{z}-\bar{\mathbf{x}}_0\|^3. \end{aligned}$$

Proof. The proof is based on mathematical induction. The base case $l = 0$ corresponds to $f(\bar{\mathbf{x}}_0) = \psi_0(\mathbf{z}_0)$, which follows from the definition of $\psi_0(\mathbf{z})$. It suffices to show the inequality on the right hand side. Denote $\mathbf{x}_0 \in \mathbb{R}^d$ as the initial iterate in Algorithm 2, $\bar{\mathbf{x}}_0$ is the output returned by Algorithm 2 and $\bar{\mathbf{s}}_0^m$ as a global minimizer of $m(\mathbf{s}, \mathbf{x}_0, \sigma_0^{ASAS})$ over $\mathbb{R}^d$. We also note that for each σ_i in Algorithm 2, $\sigma_i \geq \sigma_{\min}$ and thus $\mathcal{L}(\mathbf{x}_0, \sigma_i) \subseteq \mathcal{L}(\mathbf{x}_0, \sigma_{\min})$. Then, noting $\bar{\mathbf{x}}_0 = \mathbf{x}_0 + \bar{\mathbf{s}}_0$, by (8) one has

$$\|\mathbf{x}_0 + \bar{\mathbf{s}}_0 - \mathbf{x}^\star\| \leq D \quad \text{and} \quad \|\mathbf{x}_0 + \bar{\mathbf{s}}_0^m - \mathbf{x}^\star\| \leq D. \tag{18}$$

Furthermore, by the criterion of successful iteration in Algorithm 2,

$$f(\bar{\mathbf{x}}_0) \leq m(\bar{\mathbf{s}}_0, \mathbf{x}_0, \sigma_0^{ASAS}) = (m(\bar{\mathbf{s}}_0, \mathbf{x}_0, \sigma_0^{ASAS}) - m(\bar{\mathbf{s}}_0^m, \mathbf{x}_0, \sigma_0^{ASAS})) + m(\bar{\mathbf{s}}_0^m, \mathbf{x}_0, \sigma_0^{ASAS}).$$

Since $\|\nabla^2 f(\mathbf{x}_i) - \tilde{H}(\mathbf{x}_i)\| \leq \epsilon_i$ for all i, and f is convex, we have (13) holds and $H(\mathbf{x}_i) \succeq 0$. Besides, we note that $\nabla^2\left(\|\mathbf{s}\|^3\right) = 3\left(\|\mathbf{s}\| \cdot \mathbb{I} + \mathbf{s}\mathbf{s}^\top\right) \succeq 0$. Therefore, $m(\mathbf{s}, \mathbf{x}_0, \sigma_0^{ASAS})$ is convex and we have

$$\begin{aligned} & m(\bar{\mathbf{s}}_0, \mathbf{x}_0, \sigma_0^{ASAS}) - m(\bar{\mathbf{s}}_0^m, \mathbf{x}_0, \sigma_0^{ASAS}) \\ \leq\ & (\nabla f(\mathbf{x}_0) + H(\mathbf{x}_0)\bar{\mathbf{s}}_0 + \sigma_0^{ASAS}\|\bar{\mathbf{s}}_0\| \cdot \bar{\mathbf{s}}_0)^\top(\bar{\mathbf{s}}_0 - \bar{\mathbf{s}}_0^m) \\ \leq\ & \|\nabla f(\mathbf{x}_0) + H(\mathbf{x}_0)\bar{\mathbf{s}}_0 + \sigma_0^{ASAS}\|\bar{\mathbf{s}}_0\| \cdot \bar{\mathbf{s}}_0\| \cdot \|\bar{\mathbf{s}}_0 - \bar{\mathbf{s}}_0^m\| \\ \overset{(15)}{\leq}\ & \kappa_\theta\|\bar{\mathbf{s}}_0\|^2\|\bar{\mathbf{s}}_0 - \bar{\mathbf{s}}_0^m\| \\ \leq\ & \kappa_\theta\|\bar{\mathbf{s}}_0 + \mathbf{x}_0 - \mathbf{x}^\star - (\mathbf{x}_0 - \mathbf{x}^\star)\|^2\|\bar{\mathbf{s}}_0 + \mathbf{x}_0 - \mathbf{x}^\star - (\bar{\mathbf{s}}_0^m + \mathbf{x}_0 - \mathbf{x}^\star)\| \\ \overset{(18)\,(8)}{\leq}\ & 8\kappa_\theta D^3. \end{aligned}$$

On the other hand, we also have

$$\begin{aligned} & m(\bar{\mathbf{s}}_0^m, \mathbf{x}_0, \sigma_0^{ASAS}) \\ =\ & f(\mathbf{x}_0) + (\bar{\mathbf{s}}_0^m)^\top\nabla f(\mathbf{x}_0) + \frac{1}{2}(\bar{\mathbf{s}}_0^m)^\top H(\mathbf{x}_0)\bar{\mathbf{s}}_0^m + \frac{1}{3}\sigma_0^{ASAS}\|\bar{\mathbf{s}}_0^m\|^3 \\ \leq\ & f(\mathbf{x}_0) + (\mathbf{z}-\mathbf{x}_0)^\top\nabla f(\mathbf{x}_0) + \frac{1}{2}(\mathbf{z}-\mathbf{x}_0)^\top\nabla^2 f(\mathbf{x}_0)(\mathbf{z}-\mathbf{x}_0) + \frac{\epsilon_0}{2}\|\mathbf{z}-\mathbf{x}_0\|^2 + \frac{\sigma_0^{ASAS}}{3}\|\mathbf{z}-\mathbf{x}_0\|^3 \\ \leq\ & f(\mathbf{z}) + \frac{\bar{L}}{2}\|\mathbf{z}-\mathbf{x}_0\|^2 + \frac{\epsilon_0}{2}\|\mathbf{z}-\mathbf{x}_0\|^2 + \frac{\sigma_0^{ASAS}}{3}\|\mathbf{z}-\mathbf{x}_0\|^3 \\ \leq\ & f(\mathbf{z}) + \frac{\bar{L}+\epsilon_0}{2}\|\mathbf{z}-\mathbf{x}_0\|^2 + \frac{\bar{\sigma}_1^P}{3}\|\mathbf{z}-\mathbf{x}_0\|^3, \end{aligned}$$

where the second inequality is due to the convexity of f and (6). Therefore,

$$\psi_0(\mathbf{z}) = f(\bar{\mathbf{x}}_0)+\frac{1}{6}\varsigma_0\|\mathbf{z}-\bar{\mathbf{x}}_0\|^3 \leq f(\mathbf{z})+8\kappa_\theta D^3+\frac{\bar{L}+\epsilon_0}{2}\|\mathbf{z}-\mathbf{x}_0\|^2+\frac{\bar{\sigma}_1^P}{3}\|\mathbf{z}-\mathbf{x}_0\|^3+\frac{1}{6}\varsigma_0\|\mathbf{z}-\bar{\mathbf{x}}_0\|^3.$$

Now suppose that the theorem is proven for some $l \geq 1$. Let us consider the case of $l+1$:

$$\begin{aligned}
& \psi_{l+1}(\mathbf{z}_{l+1}) \ \leq \ \psi_{l+1}(\mathbf{z}) \\
= \ & \psi_l(\mathbf{z}) + \frac{(l+1)(l+2)}{2}[f(\bar{\mathbf{x}}_l) + (\mathbf{z}-\bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l)] + \frac{1}{6}(\varsigma_{l+1}-\varsigma_l)\|\mathbf{z}-\bar{\mathbf{x}}_0 t\|^3 \\
\leq \ & \frac{l(l+1)(l+2)}{6} f(\mathbf{z}) + 8\kappa_\theta D^3 + \frac{\bar{L}+\epsilon_0}{2}\|\mathbf{z}-\mathbf{x}_0\|^2 + \frac{\bar{\sigma}_1^P}{3}\|\mathbf{z}-\mathbf{x}_0\|^3 + \frac{\varsigma_l}{6}\|\mathbf{z}-\bar{\mathbf{x}}_0\|^3 \\
& + \frac{(l+1)(l+2)}{2}[f(\bar{\mathbf{x}}_l) + (\mathbf{z}-\bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l)] + \frac{1}{6}(\varsigma_{l+1}-\varsigma_l)\|\mathbf{z}-\bar{\mathbf{x}}_0\|^3 \\
\leq \ & \frac{(l+1)(l+2)(l+3)}{6} f(\mathbf{z}) + 8\kappa_\theta D^3 + \frac{\bar{L}+\epsilon_0}{2}\|\mathbf{z}-\mathbf{x}_0\|^2 + \frac{\bar{\sigma}_1^P}{3}\|\mathbf{z}-\mathbf{x}_0\|^3 + \frac{\varsigma_{l+1}}{6}\|\mathbf{z}-\bar{\mathbf{x}}_0\|^3,
\end{aligned}$$

where the last inequality is due to convexity of $f(\mathbf{z})$. On the other hand, noting the way that $\psi_{l+1}(\mathbf{z})$ is updated we have $\frac{(l+1)(l+2)(l+3)}{6} f(\bar{\mathbf{x}}_{l+1}) \leq \psi_{l+1}(\mathbf{z}_{l+1})$. The theorem is thus proven by induction. □

After establishing Theorem 8, the iteration complexity of Algorithm 1 readily follows.

Theorem 9 *Let ϵ be the accuracy of optimality, ϵ_i be the tolerance of sub-sampled Hessian approximation in* (16) *for iteration i, and δ_0 be the probability that inequality* (16) *fails for at least one iteration. When Algorithm 1 runs*

$$\begin{aligned}
T \ &= \ \left\lceil 1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_1^P}{\sigma_{\min}}\right)\right\rceil + \left\lceil 1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_2^P}{\sigma_{\min}}\right)\right\rceil \left\lceil \left(\frac{C^P}{\epsilon}\right)^{\frac{1}{3}}\right\rceil \\
& \quad + \left\lceil \frac{1}{\log(\gamma_3)} \log\left[\frac{8\eta^{-2}(0.5\bar{\rho}+2\kappa_\theta+L+\epsilon_0+2\bar{\sigma}_2^P+2)^3}{\eta^2\,\varsigma_0}\right] + 1\right\rceil \\
&= \ \mathcal{O}(\epsilon^{-1/3})
\end{aligned}$$

iterations (including the successful iterations to update ς), then with probability $1-\delta_0$ we have $f(\mathbf{x}_{out}) - f^\star \leq \epsilon$, where $C^P = D^3(48\kappa_\theta + 2\bar{\sigma}_1^P + \gamma_3\bar{\varsigma}^P) + 3D^2(\bar{L}+\epsilon_0)$.

Proof. Under the probability assumption of Theorem 8 and by taking $\mathbf{z} = \mathbf{x}^\star$, we have that

$$\begin{aligned}
& \frac{l(l+1)(l+2)}{6} f(\bar{\mathbf{x}}_l) \\
\leq \ & \frac{l(l+1)(l+2)}{6} f(\mathbf{x}^\star) + 8\kappa_\theta D^3 + \frac{\bar{L}+\epsilon_0}{2}\|\mathbf{x}^\star-\mathbf{x}_0\|^2 + \frac{\bar{\sigma}_1^P}{3}\|\mathbf{x}^\star-\mathbf{x}_0\|^3 + \frac{\varsigma_l}{6}\|\mathbf{x}^\star-\bar{\mathbf{x}}_0\|^3 \\
\leq \ & \frac{l(l+1)(l+2)}{6} f^\star + \left(8\kappa_\theta + \frac{\bar{\sigma}_1^P}{3} + \frac{\gamma_3\bar{\varsigma}^P}{6}\right) D^3 + \left(\frac{\bar{L}+\epsilon_0}{2}\right) D^2.
\end{aligned}$$

Rearranging the terms yields that

$$f(\bar{\mathbf{x}}_l) - f^* \leq \frac{C^P}{l(l+1)(l+2)} < \frac{C^P}{l^3}.$$

Recall that l is the count of successful iterations and $\mathcal{SC}$ is the index set of all successful iterations in Algorithm 3. Then $|\mathcal{SC}| = l < \left(\frac{C^P}{\epsilon}\right)^{1/3}$ whenever $f(\bar{x}_l) - f^* \geq \epsilon$. Therefore, by choosing $T_2 = \left\lceil 1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_2^P}{\sigma_{\min}}\right)\right\rceil \left\lceil \left(\frac{C^P}{\epsilon}\right)^{1/3}\right\rceil$ and by Lemma 6, we have $l = |\mathcal{SC}| \geq \left\lceil \left(\frac{C^P}{\epsilon}\right)^{1/3}\right\rceil$, which further implies $f(\bar{x}_l) - f^* < \epsilon$. Denote $\hat{T} = T_1 + T_2$ to be the total number of iterations that generates sub-sampled Hessian in Algorithm 1. Then combining the choice of T_2 and Lemma 5 yields that

$$\hat{T} = \left\lceil 1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_1^P}{\sigma_{\min}}\right)\right\rceil + \left\lceil 1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_2^P}{\sigma_{\min}}\right)\right\rceil \left\lceil \left(\frac{C^P}{\epsilon}\right)^{1/3}\right\rceil = \mathcal{O}(\epsilon^{-1/3})$$

To ensure an overall accumulative success probability of $1 - \delta_0$ for the entire $\hat{T}$ iterations, the per-iteration failure probability is set as $1 - \sqrt[\hat{T}]{1 - \delta_0} = \mathcal{O}(\delta_0/\hat{T}) = \mathcal{O}(\delta_0 \epsilon^{1/3})$; see Xu et al. (2020b) for more details. Therefore, by setting $\hat{\epsilon} = \epsilon_i$ and $\delta = \delta_0 \epsilon^{1/3}$ in Lemma 3 (or Lemma 4) and we have that $\|\nabla^2 f(\mathbf{x}_i) - \tilde{H}(\mathbf{x}_i)\| \leq \epsilon_i$ for all $i \leq T_1 + T_2$ with probability $1 - \delta_0$. As a result, the probability assumption in Theorem 8 is satisfied, and the conclusion follows from the choice of $\hat{T}$ and Lemma 7.

□

5. Worst-Case Iteration Complexity

In this section, we consider the case where the accuracy requirement of the sub-sampled Hessian is not satisfied, and assume that each component function f_i in f of problem (1) is convex. For any $H(\mathbf{x})$ constructed in Algorithm 2 and Algorithm 3, noting that ϵ_0 is the upper bound of the tolerance of all Hessian approximations in the algorithms, it holds that

$$H(\mathbf{x}) \succeq 0 \quad \text{and} \quad \|H(\mathbf{x})\| \leq \frac{1}{n|\mathcal{S}|} \sum_{j \in \mathcal{S}} \frac{1}{p_j} \|\nabla^2 f_j(\mathbf{x}_i)\| + \epsilon_0 \overset{(6)}{\leq} L + \epsilon_0. \tag{19}$$

In the following, to highlight the flow of our analysis, we shall present the lemmas key to our analysis but relegate their proofs to the appendix.

Lemma 10 *Suppose $\|\nabla f(\mathbf{x}_i)\|^2 > \epsilon$ in each iteration i of Algorithm 2. Denoting*

$$\bar{\sigma}_1^W = \max\left\{\sigma_0,\ \frac{3\gamma_2 L(4L + \epsilon_0)}{(1 - \kappa_\theta)\sqrt{\epsilon}}\right\} > 0,$$

we have

$$T_1 \leq \left\lceil 1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_1^W}{\sigma_{\min}}\right)\right\rceil.$$

Lemma 11 *Suppose $\|\nabla f(\mathbf{x}_j)\|^2 > \epsilon$ in each iteration j of Algorithm 3. Denoting*

$$\bar{\sigma}_2^W = \max\left\{\bar{\sigma}_1^W, \gamma_2 \frac{(3L + 2\epsilon_0)(2L + \epsilon_0) + 2\sqrt{\epsilon}(1 - \kappa_\theta)(\kappa_\theta + \eta) + (2L + \epsilon_0)\sqrt{(3L + 2\epsilon_0)^2 + \sqrt{\epsilon}(1 - \kappa_\theta)(\kappa_\theta + \eta)}}{2\sqrt{\epsilon}(1 - \kappa_\theta)}\right\}. \tag{20}$$

we have

$$T_2 \leq \left\lceil 1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_2^W}{\sigma_{\min}}\right) \right\rceil |\mathcal{SC}|.$$

Now we are ready to estimate an upper bound of T_3: the total counts of successfully updating $\varsigma > 0$.

Lemma 12 *Suppose in each iteration j of Algorithm 3, we have $\|\nabla f(\mathbf{x}_j)\|^2 > \epsilon$ for all $0 \leq j \leq T_2$. Then inequality* (17) *holds if*

$$\varsigma_l \geq \bar{\varsigma}^W := \frac{8}{\eta^2}\left((2L+\epsilon_0)\cdot\frac{(L+\epsilon_0)+\sqrt{(L+\epsilon_0)^2+4\bar{\sigma}_2^W\sqrt{\epsilon}(1-\kappa_\theta)}}{2\sqrt{\epsilon}(1-\kappa_\theta)}+\bar{\sigma}_2^W+\kappa_\theta\right)^3, \quad (21)$$

where $\bar{\sigma}_2^W$ is defined in (20)*, and it further implies that*

$$T_3 \leq \left\lfloor \frac{1}{\log(\gamma_3)} \log\left[\frac{8}{\eta^2\varsigma_0}\left((2L+\epsilon_0)\cdot\frac{(L+\epsilon_0)+\sqrt{(L+\epsilon_0)^2+4\bar{\sigma}_2^W\sqrt{\epsilon}(1-\kappa_\theta)}}{2\sqrt{\epsilon}(1-\kappa_\theta)}+\bar{\sigma}_2^W+\kappa_\theta\right)^3\right] + 1 \right\rfloor$$

In the rest of this section, we refer the combined number of iterations of the two subroutines (Algorithm 2 and Algorithm 3) as the iteration count for Algorithm 1.

Theorem 13 *Suppose that every component function f_i in f of problem* (1) *is convex and in each iteration i of Algorithm 1, we have $\|\nabla f(\mathbf{x}_j)\|^2 > \epsilon$ for all j. Then the sequence $\{\bar{\mathbf{x}}_l,\ l = 0, 1, \ldots\}$ generated by Algorithm 3 satisfies*

$$\begin{aligned} & \frac{l(l+1)(l+2)}{6} f(\bar{\mathbf{x}}_l) \leq \psi_l(\mathbf{z}_l) \leq \psi_l(\mathbf{z}) \\ \leq\ & \frac{l(l+1)(l+2)}{6} f(\mathbf{z}) + 8\kappa_\theta D^3 + \frac{L+\epsilon_0}{2}\|\mathbf{z}-\mathbf{x}_0\|^2 + \frac{\bar{\sigma}_1^W}{3}\|\mathbf{z}-\mathbf{x}_0\|^3 + \frac{\varsigma_l}{6}\|\mathbf{z}-\bar{\mathbf{x}}_0\|^3. \end{aligned}$$

Proof. The proof is almost identical to that of Theorem 8 (which is based on mathematical induction) except the following estimation on $m(\bar{\mathbf{s}}_0^m, \mathbf{x}_0, \sigma_0^{ASAS})$, where $\bar{\mathbf{s}}_0^m$ is a global minimizer of $m(\mathbf{s}, \mathbf{x}_0, \sigma_0^{ASAS})$ over $\mathbb{R}^d$:

$$\begin{aligned} m(\bar{\mathbf{s}}_0^m, \mathbf{x}_0, \sigma_0^{ASAS}) &= f(\mathbf{x}_0) + (\bar{\mathbf{s}}_0^m)^\top \nabla f(\mathbf{x}_0) + \frac{1}{2}(\bar{\mathbf{s}}_0^m)^\top H(\mathbf{x}_0)\bar{\mathbf{s}}_0^m + \frac{1}{3}\sigma_0^{ASAS}\|\bar{\mathbf{s}}_0^m\|^3 \\ &\overset{(19)}{\leq} f(\mathbf{x}_0) + (\mathbf{z}-\mathbf{x}_0)^\top \nabla f(\mathbf{x}_0) + \frac{1}{2}(L+\epsilon_0)\|\mathbf{z}-\mathbf{x}_0\|^2 + \frac{\sigma_0^{ASAS}}{3}\|\mathbf{z}-\mathbf{x}_0\|^3 \\ &\leq f(\mathbf{z}) + \frac{L+\epsilon_0}{2}\|\mathbf{z}-\mathbf{x}_0\|^2 + \frac{\bar{\sigma}_1^W}{3}\|\mathbf{z}-\mathbf{x}_0\|^3, \end{aligned}$$

where the second inequality is due to the convexity of f. Then, by replacing the estimation of $m(\bar{\mathbf{s}}_0^m, \mathbf{x}_0, \sigma_0^{ASAS})$ with the inequality above, the conclusion readily follows. □

After establishing Theorem 13 and denoting

$$\bar{\sigma}^W := \max\left\{\sigma_0, \frac{3\gamma_2 L(4L+\epsilon_0)}{(1-\kappa_\theta)}, \gamma_2\frac{(3L+2\epsilon_0)(2L+\epsilon_0)+2(1-\kappa_\theta)(\kappa_\theta+\eta)+(2L+\epsilon_0)\sqrt{(3L+2\epsilon_0)^2+(1-\kappa_\theta)(\kappa_\theta+\eta)}}{2(1-\kappa_\theta)}\right\}, \quad (22)$$

the iteration complexity of Algorithm 1 readily follows.

Theorem 14 *Suppose every component function f_i in f of problem (1) is convex, and let $0 < \epsilon < 1$ sufficiently small. The Algorithm 1 returns a solution $\mathbf{x}_{out}$ such that either $\|\nabla f(\mathbf{x}_{out})\|^2 \leq \epsilon$ or $f(\mathbf{x}_{out}) - f^* \leq \epsilon$, at an iteration no more than*

$$\begin{aligned} T &\leq \left\lceil 1 + \frac{2}{\log(\gamma_1^W)} \log\left(\frac{\bar{\sigma}^W}{\sigma_{\min}} \epsilon^{-\frac{1}{2}}\right)\right\rceil + \left\lceil 1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}^W}{\sigma_{\min}} \epsilon^{-\frac{1}{2}}\right)\right\rceil \left\lceil (C^W)^{\frac{1}{3}} \cdot \epsilon^{-\frac{5}{6}} \right\rceil \\ &\quad + \left\lceil \frac{1}{\log(\gamma_3)} \log\left[\left((2L+\epsilon_0) \cdot \frac{(L+\epsilon_0) + \sqrt{(L+\epsilon_0)^2 + 4\bar{\sigma}^W(1-\kappa_\theta)}}{2(1-\kappa_\theta)} + \bar{\sigma}^W + \kappa_\theta\right)^3 \frac{8D^3}{\eta^2 \varsigma_0} \epsilon^{-\frac{3}{2}}\right] + 1 \right\rceil \\ &= \mathcal{O}(\epsilon^{-5/6} \log(\epsilon^{-1})), \end{aligned}$$

where

$$C^W = 12\kappa_\theta D^3 + 3(L+\epsilon_0)D^2 + 2\bar{\sigma}^W D^3 + \frac{8D^3}{\eta^2}\left((2L+\epsilon_0) \cdot \frac{(L+\epsilon_0) + \sqrt{(L+\epsilon_0)^2 + 4\bar{\sigma}^W(1-\kappa_\theta)}}{2(1-\kappa_\theta)} + \bar{\sigma}^W + \kappa_\theta\right)^3 \tag{23}$$

and $\bar{\sigma}^W$ is defined in (22).

Proof. Suppose Algorithm 1 does not stop early, i.e., we have $\|\nabla f(\mathbf{x}_j)\|^2 > \epsilon$ in every iteration j. Recall that $l = 0, 1, \ldots$ is the count of successful iterations in Algorithm 3. Applying the inequality in Theorem 13 with $\mathbf{z} = \mathbf{x}^\star$ we have

$$\frac{l(l+1)(l+2)}{6}(f(\bar{\mathbf{x}}_l) - f(\mathbf{x}^\star)) \leq 8\kappa_\theta D^3 + \left(\frac{L+\epsilon_0}{2}\right) D^2 + \frac{\bar{\sigma}_1^W}{3} D^3 + \frac{\varsigma_l}{6} D^3$$

Note that $\varsigma_l \leq \gamma_3 \bar{\varsigma}^W$ and $\bar{\varsigma}^W$ has the magnitude of $\epsilon^{-\frac{3}{2}}$ in (21). In addition, $\bar{\sigma}_1^W$ that is defined in Lemma 10, is also dependent on ϵ. The above inequality implies that

$$f(\bar{\mathbf{x}}_l) - f(\mathbf{x}^\star) \leq \frac{C^W}{l(l+1)(l+2)} \cdot \epsilon^{-\frac{3}{2}} < \frac{C^W}{l^3} \cdot \epsilon^{-\frac{3}{2}} \tag{24}$$

with C^W defined in (23). Recall that $\mathcal{SC}$ is the index set of all successful iterations in Algorithm 3. Then $|\mathcal{SC}| = l < \left(\frac{C^W}{\epsilon^{5/2}}\right)^{1/3}$ whenever $f(\bar{x}_l) - f^* \geq \epsilon$. Therefore, by choosing $T_2 = \left\lceil 1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_2^W}{\sigma_{\min}}\right)\right\rceil \left\lceil \left(\frac{C^W}{\epsilon^{5/2}}\right)^{1/3} \right\rceil$ and Lemma 11, we must have $l = |\mathcal{SC}| \geq \left\lceil \left(\frac{C^W}{\epsilon^{5/2}}\right)^{1/3} \right\rceil$, which further implies $f(\bar{x}_l) - f^* < \epsilon$. Finally, the upper bound on T follows by combining this result with lemmas 10 and 12. □

To conclude this section, we remark that if we adopt a stronger early stop criterion of $\|\nabla f(\mathbf{x}_{j+1})\| \leq \epsilon$ in Algorithm 1, then the iteration bound in Theorem 14 will change to $\mathcal{O}(\epsilon^{-4/3} \log(\epsilon^{-1/2}))$. This is because in this case, the factor $\sqrt{\epsilon}$ in the bound of $\bar{\varsigma}^W$ from (21) is replaced by ϵ. As a result, the quantity $\frac{C^W}{l^3} \cdot \epsilon^{-\frac{3}{2}}$ in (24) is adapted to $\frac{C^W}{l^3} \cdot \epsilon^{-3}$. Then we can let $T_2 = \left\lceil 1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_2^W}{\sigma_{\min}}\right)\right\rceil \left\lceil \left(\frac{C^W}{\epsilon^4}\right)^{1/3} \right\rceil$ and guarantee $l \geq \lceil (C^W)^{\frac{1}{3}} \epsilon^{-\frac{4}{3}} \rceil$ by Lemma 10, which further implies $f(\bar{x}_l) - f^* < \epsilon$. The iteration bound of $\mathcal{O}(\epsilon^{-4/3} \log(\epsilon^{-1/2}))$ follows from this result, lemmas 10 and 12.

Table 1: The Statistics of Eight LIBSVM Datasets

Name	Instances No.	Features No.	Processing
SUSY	5,000,000	18	Done by Baldi et al. (2014)
covtype	581,012	54	Transformed from multiclass by Collobert et al. (2002).
phishing	11,055	68	Binary encoding and length-normalized
w8a	49,749	300	Rescaled to a unit vector
gisette	7,000	5,000	Feature-wisely rescaled within $[-1, 1]$
rcv1	20,242	47,236	Only training data used
real-sim	72,309	20,958	Vikas Sindhwani for the SVMlin project

6. Numerical Experiments

We shall demonstrate the efficacy of the proposed method by presenting some computational results on different genres of real data. Experimental results on regularized logistic regression confirm that our algorithm is suitable for solving large-scale statistical learning problems and at least competitive with other algorithms. In addition, all eight data sets are selected from the LIBSVM collection[1] in which their statistics are summarized in Table 1, and all algorithms are implemented using Python 3.5 on a MacBook Pro running with Mac OS High Sierra 10.13.6 and 16GB memory.

Problem. Given a collection of data samples $\{(\mathbf{w}_i, y_i)\}_{i=1}^n$ in which $y_i \in \{-1, 1\}$, the model of regularized logistic regression is given by

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i \mathbf{w}_i^\top \mathbf{x}}\right) + \left(\frac{\lambda}{2}\right) \|\mathbf{x}\|_2 . \tag{25}$$

where the regularization term $\|\cdot\|_2$ promotes smoothness and $\lambda > 0$ balances smoothness with goodness-of-fit and generalization and is chosen by five-fold cross validation.

Experimental setting. We implement Algorithm 1 with $\eta = 0.1$, $\gamma_1 = \gamma_2 = \gamma_3 = 2$, $\sigma_{\min} = 10^{-16}$, $\sigma_0 = 1$ and $\kappa_\theta = 0.1$, denoted as SACR, in a hybrid manner. Specifically, given that SACR contains two phases we implement SACR with these two phases at the beginning and stop the second phase when the iterate is relatively close to the optimal solution, and then switch to subsampled cubic regularization (SCR) method. This is because that we observe that the first phase mainly contributes to the local convergence of SACR while the second phase may hurt it. In fact, when the iterate is close enough to an optimal solution, the first phase reduces to the Newton method, hence admitting a local quadratic convergence rate. In our experiment, we stop the second phase when $|f(\mathbf{x}_{i+1}) - f(\mathbf{x}_i)|/|f(\mathbf{x}_i)| \leq 10^{-1}$ and the final stopping criterion as $\|\nabla f(x)\| \leq 10^{-7}$.

Furthermore, since the accelerated convergence of our algorithm is global, to observe the effect of acceleration, we need to set the initial solution far away from the local convergence region. In this case, we randomly generate the starting point from a Gaussian random variable with zero mean and a large variance. The sample size is chosen inversely proportional to the square norm of the gradient (cf. Lemmas 3 and 4) with proportional

1. The LIBSVM collection is available at https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets

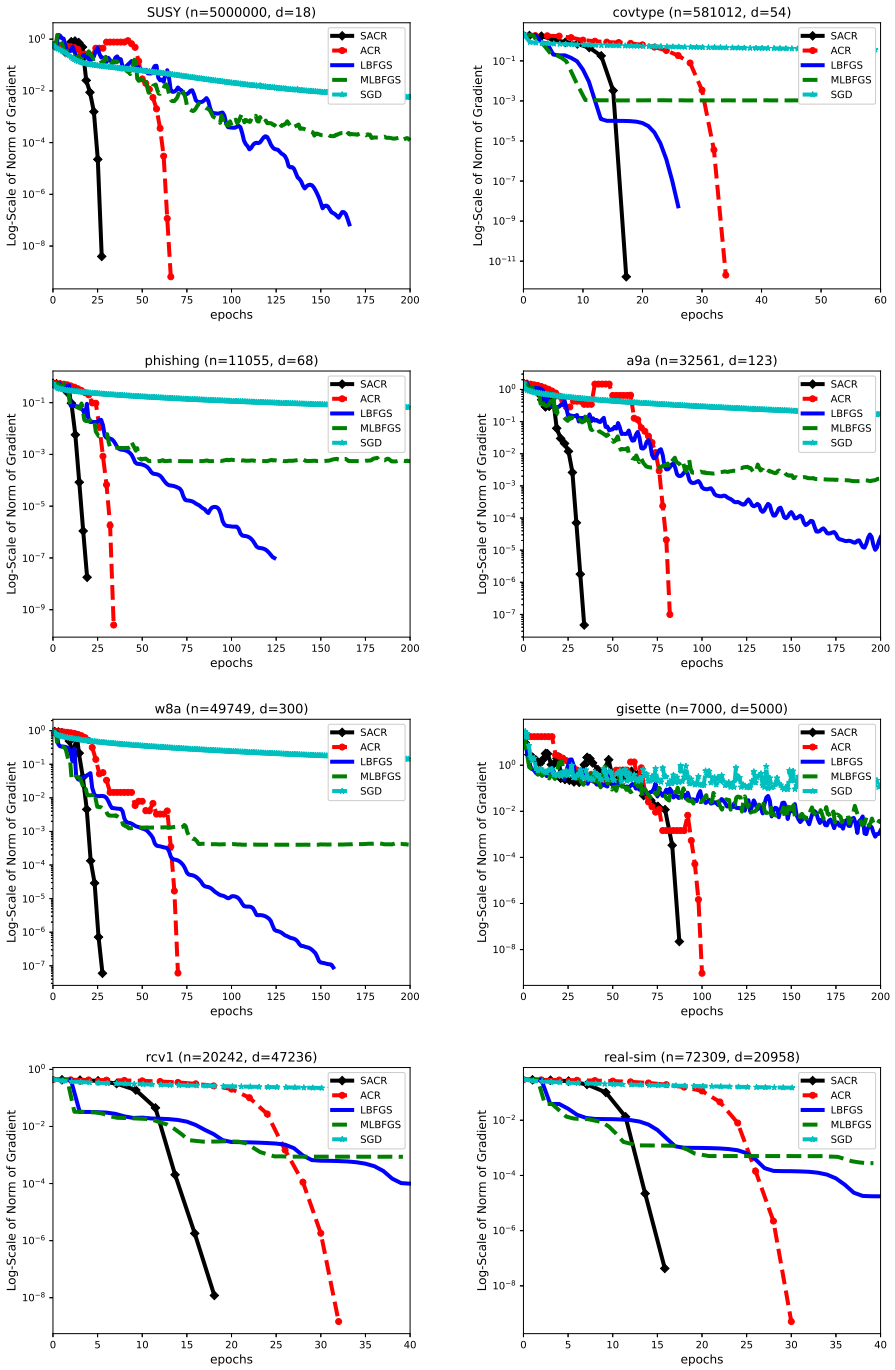

Figure 1: Performance of our algorithm and four state-of-the-art algorithms without sub-sampled Hessian information on eight datasets with the log-scale of the norm of gradient vs. number of epochs.

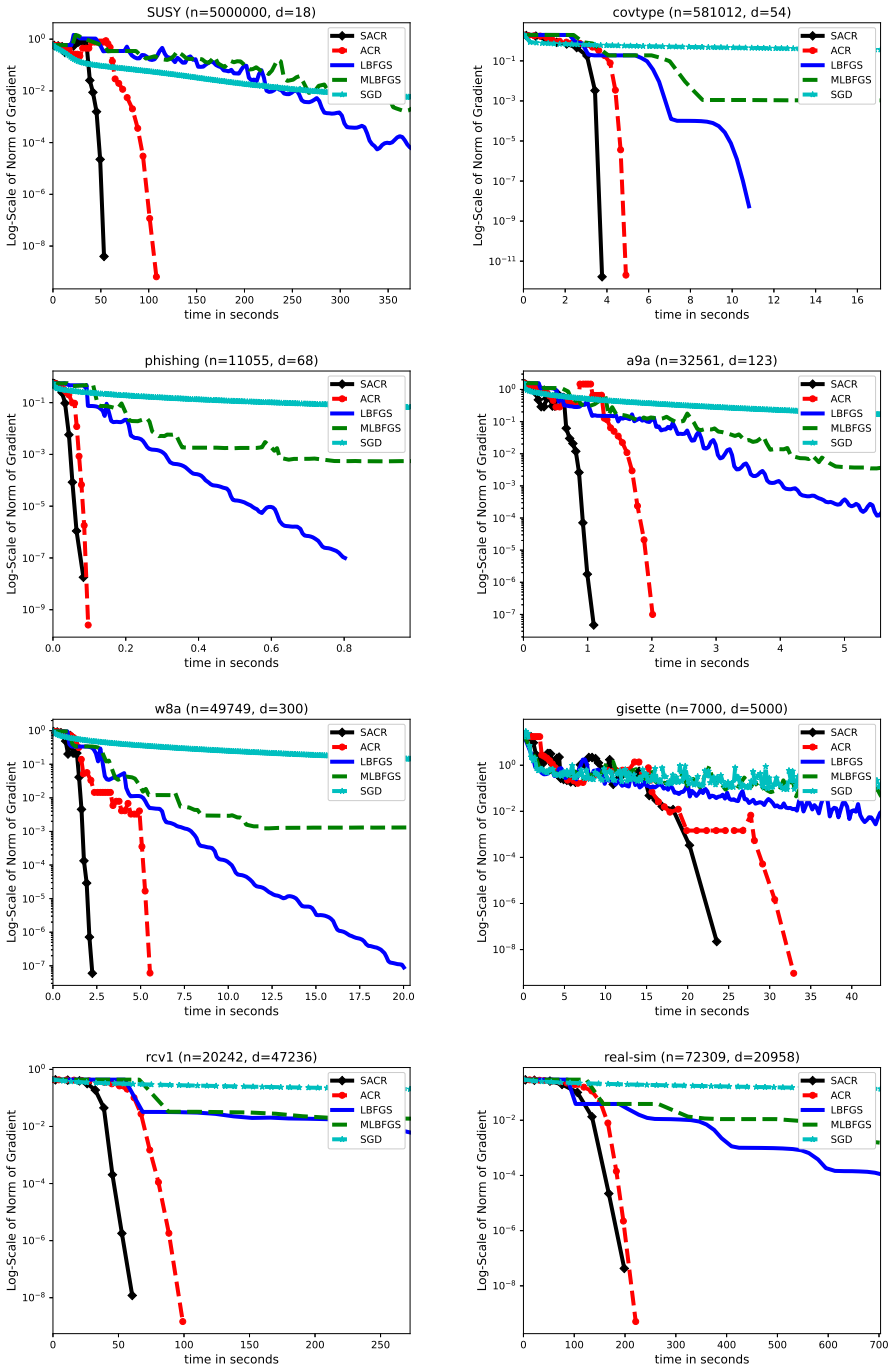

Figure 2: Performance of our algorithm and four state-of-the-art algorithms without sub-sampled Hessian information on eight datasets with the log-scale of the norm of gradient vs. time.

ratio $0.2\log(100d)$ and are bounded both below and above by some constants. The lower bound and the upper bound are tuned for different datasets. In addition, we require the lower bound of the sample size decreases to a smaller value after we switching SACR to SCR in the local region of the optimal solution.

Finally, we use the log-scale of norm of gradient as the function of number of epochs and run time as the metric in our experiment. In particular, one epoch is counted when a full batch size (i.e. n times) of the gradient or Hessian of the component functions is queried. Since the sample size in sub-sampling algorithms is less than n, one epoch is likely to be consumed by the queries from several iterations.

Subproblem solving. The generalized conjugate gradient method with Lanczos process is applied to approximately solve the cubic regularized subproblem. More specifically, the convex cubic polynomial in the subproblem is minimized over a Krylov subspace, which is defined by the gradient and Hessian of f at $\mathbf{x}_i$ and given by

$$\mathcal{K} := \mathsf{Span}\left\{\nabla f(\mathbf{x}_i),\ \nabla^2 f(\mathbf{x}_i)\nabla f(\mathbf{x}_i),\ \left(\nabla^2 f(\mathbf{x}_i)\right)^2 \nabla f(\mathbf{x}_i),\ \ldots\right\},$$

Note that the Krylov subspace $\mathcal{K}$ gradually swells with very cheap computational cost since each orthogonal basis is created by performing a single matrix-vector product. Furthermore, the minimization of cubic polynomial over the Krylov subspace only requires factorizing a tri-diagonal matrix at the $O(d)$ expense. Finally, we set the stopping criterion for subproblem solving as (15) with $\kappa_\theta = 0.1$.

6.1 Comparision with four state-of-the-art algorithms without sub-sampled Hessian information

In the first experiment, we compare our algorithm to four baseline algorithms, including the deterministic counterpart of our algorithm Jiang et al. (2020), denoted as ACR, the limited memory BFGS method, denoted as LBFGS, the minibatch variant of LBFGS with the batch size $n/2$, denoted as MLBFGS, and the minibatch variant of stochastic gradient descent with the batch size $n/10$, denoted as SGD. Note that only the gradient is sampled in MLBFGS and SGD, while the Hessian instead of the gradient is subsampled in our algorithm. For LBFGS and MLBFGS implementations, we set an initial matrix as identity matrix and the line search criterion with strong Wolfe condition. The ratio for measuring the progress is set as 0.9, the maximum number of line search is set as 5 and the memory size is set as 30. Additionally, we excluded the sub-sampled Newton method since its global convergence is unknown in general and, when the iterate is close enough, our algorithm turns out to be the same as the sub-sampled Newton method since the cubic regularization term will become very small.

The results on eight datasets are presented in Figures 1-2. We observe that SACR outperforms other algorithms in most of the datasets despite the competitive performance of other algorithms at the initial stage. In particular, both SACR and ACR can attain the solution with high accuracy while LBFGS, MLBFGS and SGD can not. We observe the curve of LBFGS lies between that of SGD and ACR, and it behaves more like ACR for low-dimensional dataset (i.e., SUSY and covtype). This is probably due to that LBFGS can be viewed as an interplay between the first order and the second order method, and

it exhibits superlinear convergence thanks to certain geometrical regularity (e.g., restricted strongly convexity) that intuitively exists for low-dimensional problem with high probability in real applications. Also, SACR is more efficient than ACR due to the usage of sub-sampling techniques. When the dimension of the dataset becomes larger, standard second-order methods suffer from the storing and computing the inverse of the Hessian as the dimension increases, while our algorithm remains efficient in most of these datasets. This is not surprising since the subproblem solving depends on the generalized conjugate gradient method. For high-dimensional problems, storing the Hessian appears to be a critical issue which requires further exploration. To this end, the competitive performance demonstrates that our algorithm has a great potential to achieve practical performance on the large-scale problems.

6.2 Comparision with three types of sub-sampled cubic regularized algorithms

In the second experiment, we compare our algorithm to three types of sub-sampled cubic regularized algorithms including the non-accelerated sub-sampled cubic regularized (SCR) algorithm that is used in the phase I of SACR, a variant of SCR in (Kohler and Lucchi, 2017) denoted as SCR-KL, a dynamic inexact Hessian variant of SCR in (Bellavia et al., 2021) denoted as SCR-BGM. In the implementation, the the sample size for the approximation in SCR-KL is determined by the previous stepsize instead of the current stepsize used in the theory (Kohler and Lucchi, 2017) with an adaptive rule[1]. While the parameters of SCR-BGM strictly follows the setting for testing the real datasets in (Bellavia et al., 2021). In addition, we impose a lower bound as well as an upper bound of the sample size for all the algorithms, and these bounds are tuned for different datasets. In particular, the lower bound is uniformly set to be $0.01 \cdot n$ for all datasets, while the upper bound is set to be $0.2 \cdot n$ for 7 datasets except the full batch size n is used for the dataset "gisette".

We provide the corresponding numerical results on eight datasets in Figures 3, where the log-scale of the norm of gradient vs. number of epochs is provided. We can see that SACR outperforms other sub-sampled cubic regularized algorithms in all the datasets, while all the tested algorithms have similar convergence behavior after the iteration points entering the local region of the optimal solution. This indeed indicates that our technique really accelerates the algorithm in finding such local region and yields a faster convergence rate.

7. Concluding Remarks

The theoretical properties of subsampled Hessian Newton-type methods have recently received a lot of attention, but their acceleration has not been well studied in the literature. In this paper, we focus on the sum-of-nonconvex problem and propose a novel way to accelerate adaptive cubic regularization of Newton's method with either *uniform* or *non-uniform* sub-sampled Hessians. Our new algorithm achieves the global iteration complexity of $O(\epsilon^{-1/3})$ with high probability, which matches that of the original accelerated cubic regularization methods (Jiang et al., 2020) using the *full* Hessian information. In the worst case scenario, we demonstrate that our algorithm still achieves an $O(\epsilon^{-5/6}\log(\epsilon^{-1}))$ iteration complexity bound. The proof techniques are new to our knowledge and can be of independent interets.

1. https://github.com/dalab/subsampled_cubic_regularization

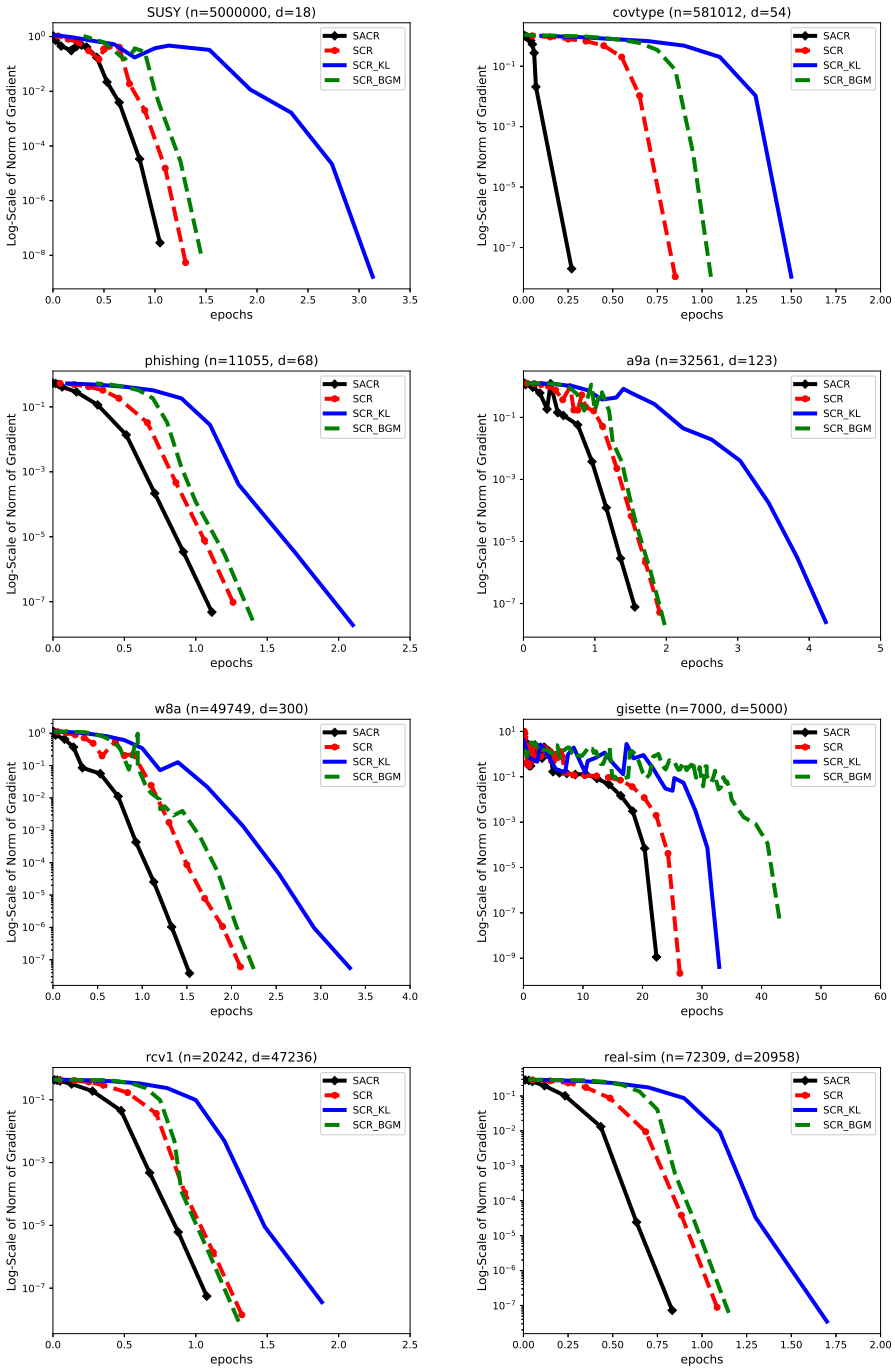

Figure 3: Performance of our algorithm and three sub-sampled cubic regularized algorithms on eight datasets with the log-scale of the norm of gradient vs. number of epochs.

Our empirical evaluation show that the new algorithm studied in this paper is generally more efficient than its deterministic counterpart, LBFGS, mini-batch LBFGS, and SGD, on regularized logistic regression problems for real datasets.

Acknowledgments

Xi Chen and Bo Jiang are co-corresponding authors. We would like to thank the three anonymous referees for their insightful comments. Bo Jiang's research is supported by the National Natural Science Foundation of China (Grants 72171141, 72150001 and 11831002), and Program for Innovative Research Team of Shanghai University of Finance and Economics.

References

N. Agarwal, B. Bullins, and E. Hazan. Second-order stochastic optimization for machine learning in linear time. *The Journal of Machine Learning Research*, 18(1):4148–4187, 2017.

Z. Allen-Zhu. Katyusha: the first direct acceleration of stochastic gradient methods. In *STOC*, pages 1200–1205. ACM, 2017.

Z. Allen-Zhu and Y. Yuan. Improved SVRG for non-strongly-convex or sum-of-non-convex objectives. In *ICML*, pages 1080–1089, 2016.

Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. In *NIPS*, pages 2675–2686, 2018.

Y. Arjevani, O. Shamir, and R. Shiff. Oracle complexity of second-order methods for smooth convex optimization. *Mathematical Programming*, 178(1-2):327–360, 2019.

P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5:4308, 2014.

S. Bellavia, G. Gurioli, B. Morini, and P. L. Toint. Adaptive regularization algorithms with inexact evaluations for nonconvex optimization. *SIAM Journal on Optimization*, 29(4): 2881–2915, 2019.

S. Bellavia, G. Gurioli, and B. Morini. Adaptive cubic regularization methods with dynamic inexact hessian information and applications to finite-sum minimization. *IMA Journal of Numerical Analysis*, 41(1):764–799, 2021.

Stefania Bellavia and Gianmarco Gurioli. Stochastic analysis of an adaptive cubic regularization method under inexact gradient evaluations and dynamic Hessian accuracy. *Optimization*, 71(1):227–261, 2022.

A. S. Berahas, R. Bollapragada, and J. Nocedal. An investigation of Newton-sketch and subsampled Newton methods. *Optimization Methods and Software*, 35(4):661–680, 2020.

E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and P. L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163(1-2):359–368, 2017.

J. Blanchet, C. Cartis, M. Menickelly, and K. Scheinberg. Convergence rate analysis of a stochastic trust-region method via supermartingales. *INFORMS Journal on Optimization*, 1(2):92–119, 2019.

R. Bollapragada, R. H. Byrd, and J. Nocedal. Exact and inexact subsampled Newton methods for optimization. *IMA Journal of Numerical Analysis*, 39(2):545–578, 2019.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

S. Bubeck, Q. Jiang, Y. T. Lee, Y. Li, and A. Sidford. Near-optimal method for highly smooth convex optimization. In *COLT*, pages 492–507, 2019.

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

R.H Byrd, S.L Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

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

C. Cartis, N. I. M. 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, 2011a.

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

R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of SVMs for very large scale problems. In *NIPS*, pages 633–640, 2002.

G. Cormode and C. Dickens. Iterative Hessian sketch in input sparsity time. *ArXiv Preprint:1910.14166*, 2019.

N. Doikov and P. Richtárik. Randomized block cubic Newton method. In *International Conference on Machine Learning*, pages 1290–1298. PMLR, 2018.

P. Drineas and M. W. Mahoney. Lectures on randomized numerical linear algebra. *The Mathematics of Data*, 25:1, 2018.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

M. A. Erdogdu and A. Montanari. Convergence rates of sub-sampled newton methods. In *NIPS*, pages 3052–3060. MIT Press, 2015.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

R. Frostig, R. Ge, S. Kakade, and A. Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *ICML*, pages 2540–2548, 2015.

D. Garber and E. Hazan. Fast and simple PCA via convex optimization. *ArXiv Preprint: 1509.05647*, 2015.

A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, and C. A. Uribe. Optimal tensor methods in smooth convex and uniformly convex optimization. In *COLT*, pages 1374–1391, 2019.

S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

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

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford. Accelerating stochastic gradient descent for least squares regression. In *COLT*, pages 545–604, 2018.

B. Jiang, T. Lin, and S. Zhang. A unified adaptive tensor approximation scheme to accelerate composite convex optimization. *SIAM Journal on Optimization*, 30(4):2897–2926, 2020.

B. Jiang, H. Wang, and S. Zhang. An optimal high-order tensor method for convex optimization. *Mathematics of Operations Research*, 46(4):1390–1412, 2021.

J. M. Kohler and A. Lucchi. Subsampled cubic regularization for non-convex optimization. In *ICML*, pages 1895–1904, 2017.

B. Kulis. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4): 287–364, 2013.

S. B. Kylasa, F. Roosta-Khorasani, M. W. Mahoney, and A. Grama. GPU accelerated subsampled Newton's method. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 702–710. SIAM, 2019.

G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397, 2012.

X. Li, S. Wang, and Z. Zhang. Do subsampled Newton methods work for high-dimensional data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4723–4730, 2020.

X. Liu, C-J. Hsieh, J. D. Lee, and Y. Sun. An inexact subsampled proximal Newton-type method for large-scale machine learning. *ArXiv Preprint: 1708.08552*, 2017.

D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*, volume 2. Springer, 1984.

L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean. Boosting algorithms as gradient descent. In *NIPS*, pages 512–518, 2000.

R. D. C. Monteiro and B. F. Svaiter. Iteration-complexity of a Newton proximal extragradient method for monotone variational inequalities and inclusion problems. *SIAM Journal on Optimization*, 22(3):914–935, 2012.

R. D. C. Monteiro and B. F. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(3):1092–1125, 2013.

Yu. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, pages 543–547, 1983. (in Russian).

Yu. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media, 2004.

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

Yu. Nesterov. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming*, pages 1–27, 2019.

C. Paquette and K. Scheinberg. A stochastic line search method with expected complexity analysis. *SIAM Journal on Optimization*, 30(1):349–376, 2020.

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

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

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

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

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

N. N. Schraudolph, J. Yu, and S. Günter. A stochastic quasi-Newton method for online convex optimization. In *AISTATS*, pages 436–443, 2007.

S. Shalev-Shwartz. SDCA without duality, regularization, and individual convexity. In *ICML*, pages 747–754, 2016.

S. Shalev-Shwartz and T. Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *NIPS*, pages 378–385, 2013.

C. Song and J. Liu. Inexact proximal cubic regularized Newton methods for convex optimization. *ArXiv Preprint: 1902.02388*, 2019.

S. Sra, S. Nowozin, and S. J. Wright. *Optimization for Machine Learning*. MIT Press, 2012.

N. Tripuraneni, M. Stern, C. Jin, J. Regier, and M. I. Jordan. Stochastic cubic regularization for fast nonconvex optimization. In *NIPS*, pages 2899–2908, 2018.

X. Wang, S. Ma, D. Goldfarb, and W. Liu. Stochastic quasi-Newton methods for nonconvex stochastic optimization. *SIAM Journal on Optimization*, 27(2):927–956, 2017.

P. Xu, J. Yang, F. Roosta-Khorasani, C. Ré, and M. W. Mahoney. Sub-sampled Newton methods with non-uniform sampling. In *NIPS*, pages 3000–3008, 2016.

P. Xu, F. Roosta-Khorasani, and M. W. Mahoney. Second-order optimization for non-convex machine learning: An empirical study. In *SDM*, pages 199–207. SIAM, 2020a.

P. Xu, F. Roosta-Khorasani, and M. W. Mahoney. Newton-type methods for non-convex optimization under inexact Hessian information. *Mathematical Programming*, 184:35–70, 2020b.

Z. Yao, P. Xu, F. Roosta-Khorasani, and M. W. Mahoney. Inexact non-convex Newton-type methods. *INFORMS Journal on Optimization*, 3(2):119–226, 2021.

H. Ye, L. Luo, and Z. Zhang. Nesterov's acceleration for approximate Newton. *Journal of Machine Learning Research*, 21(142):1–37, 2020.

J. Zhang, L. Xiao, and S. Zhang. Adaptive stochastic variance reduction for subsampled newton method with cubic regularization. *INFORMS Journal on Optimization*, published online: https://doi.org/10.1287/ijoo.2021.0058, 2021.

Appendix A. Proofs in Section 4

We prove Lemma 5 and Lemma 6, which describe the relation between the total iteration numbers in Algorithm 2 and the amount of successful iterations $|\mathcal{SC}|$ in Algorithm 3.

Proof of Lemma 5: First by invoking the fact

$$f(\mathbf{x}_i+\mathbf{s}_i) = f(\mathbf{x}_i)+\mathbf{s}_i^\top \nabla f(\mathbf{x}_i)+\frac{1}{2}\mathbf{s}_i^\top \nabla^2 f(\mathbf{x}_i)\mathbf{s}_i+\int_0^1 (1-\tau)\mathbf{s}_i^\top[\nabla^2 f(\mathbf{x}_i+\tau\mathbf{s}_i)-\nabla^2 f(\mathbf{x}_i)]\mathbf{s}_i\, d\tau, \quad (26)$$

we have that

$$\begin{aligned}
& f(\mathbf{x}_i+\mathbf{s}_i) \\
= \;& f(\mathbf{x}_i)+\mathbf{s}_i^\top \nabla f(\mathbf{x}_i)+\frac{1}{2}\mathbf{s}_i^\top \nabla^2 f(\mathbf{x}_i)\mathbf{s}_i+\int_0^1 (1-\tau)\mathbf{s}_i^\top[\nabla^2 f(\mathbf{x}_i+\tau\mathbf{s}_i)-\nabla^2 f(\mathbf{x}_i)]\mathbf{s}_i\, d\tau \\
\overset{(5)}{\leq} \;& f(\mathbf{x}_i)+\mathbf{s}_i^\top \nabla f(\mathbf{x}_i)+\frac{1}{2}\mathbf{s}_i^\top \nabla^2 f(\mathbf{x}_i)\mathbf{s}_i+\frac{\bar{\rho}}{6}\|\mathbf{s}_i\|^3 \\
= \;& m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i)+\frac{1}{2}\mathbf{s}_i^\top\left(\nabla^2 f(\mathbf{x}_i)-H(\mathbf{x}_i)\right)\mathbf{s}_i+\left(\frac{\bar{\rho}}{6}-\frac{\sigma_i}{3}\right)\|\mathbf{s}_i\|^3 \\
\overset{(12)}{\leq} \;& m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i)+\epsilon_i\|\mathbf{s}_i\|^2+\left(\frac{\bar{\rho}}{6}-\frac{\sigma_i}{3}\right)\|\mathbf{s}_i\|^3.
\end{aligned}$$

Next we argue that when σ_i exceeds a certain constant, then it holds that

$$f(\mathbf{x}_i+\mathbf{s}_i)\leq m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i).$$

The analysis is conducted according to the value of $\|\mathbf{s}_i\|$ in two cases.

1. When $\|\mathbf{s}_i\|\geq 1$, we have

   $$f(\mathbf{x}_i+\mathbf{s}_i)\leq m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i)+\left(\epsilon_i+\frac{\bar{\rho}}{6}-\frac{\sigma_i}{3}\right)\|\mathbf{s}_i\|^3,$$

   which in combination with the fact that $\epsilon_i\leq\epsilon_0\leq 1$ leads to

   $$\sigma_i\geq\frac{6+\bar{\rho}}{2}\quad\Longrightarrow\quad f(\mathbf{x}_i+\mathbf{s}_i)\leq m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i).$$

2. When $\|\mathbf{s}_i\|<1$, according to Condition 3.1, it holds that

   $$\begin{aligned}
   \kappa_\theta\|\mathbf{s}_i\|>\kappa_\theta\|\mathbf{s}_i\|^2 &\geq \|\nabla f(\mathbf{x}_i)+H(\mathbf{x}_i)\mathbf{s}_i+\sigma_i\|\mathbf{s}_i\|\cdot\mathbf{s}_i\| \\
   &\geq \|\nabla f(\mathbf{x}_i)\|-\|H(\mathbf{x}_i)\|\|\mathbf{s}_i\|-\sigma_i\|\mathbf{s}_i\|^2 \\
   &\geq \|\nabla f(\mathbf{x}_i)\|-(L+\epsilon_0+\sigma_i)\|\mathbf{s}_i\|,
   \end{aligned}$$

   where the last inequality holds true since $\|\mathbf{s}_i\|<1$ and

   $$\|H(\mathbf{x}_i)\|\leq\frac{1}{n|\mathcal{S}|}\sum_{j\in\mathcal{S}}\frac{1}{p_j}\|\nabla^2 f_j(\mathbf{x}_i)\|+\epsilon_i\|\mathbb{I}\|\overset{(6)}{\leq}\frac{1}{n|\mathcal{S}|}\sum_{j\in\mathcal{S}}\frac{1}{p_j}L+\epsilon_i\leq L+\epsilon_0.$$

   This can be further rewritten as

   $$\|\mathbf{s}_i\|\geq\frac{\|\nabla f(\mathbf{x}_i)\|}{L+\epsilon_0+\sigma_i+\kappa_\theta}. \quad (27)$$

Moreover, recall that

$$f(\mathbf{x}_i+\mathbf{s}_i) \le m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i) + \left(\frac{\epsilon_i}{\|\mathbf{s}_i\|}+\frac{\bar{\rho}}{6}-\frac{\sigma_i}{3}\right)\|\mathbf{s}_i\|^3,$$

and combining the above two inequalities yields that

$$\frac{\epsilon_i(L+\epsilon_0+\sigma_i+\kappa_\theta)}{\|\nabla f(\mathbf{x}_i)\|}+\frac{\bar{\rho}}{6}-\frac{\sigma_i}{3}\le 0 \Longrightarrow f(\mathbf{x}_i+\mathbf{s}_i)\le m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i).$$

Recall in Algorithm 2 that

$$\epsilon_i = \min\left\{\frac{\|\nabla f(\mathbf{x}_i)\|}{6},\ \epsilon_0\right\} \le \frac{\|\nabla f(\mathbf{x}_i)\|}{6}, \tag{28}$$

then it suffices to show

$$\frac{L+\epsilon_0+\sigma_i+\kappa_\theta}{6}+\frac{\bar{\rho}}{6}-\frac{\sigma_i}{3}\le 0.$$

That is:

$$\sigma_i \ge L+\epsilon_0+\kappa_\theta+\bar{\rho} \Longrightarrow f(\mathbf{x}_i+\mathbf{s}_i)\le m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i).$$

In summary, we have concluded that

$$\sigma_i \ge \max\left\{\frac{6+\bar{\rho}}{2},\ L+\epsilon_0+\kappa_\theta+\bar{\rho}\right\} \Longrightarrow f(\mathbf{x}_i+\mathbf{s}_i)\le m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i), \tag{29}$$

which implies that $\sigma_i < \max\{3+0.5\bar{\rho},\ L+\epsilon_0+\kappa_\theta+\bar{\rho}\}$ for $i \le T_1-2$. Moreover,

$$\sigma_{T_1} = \sigma_{T_1-1} \le \gamma_2\sigma_{T_1-2} \le \gamma_2 \max\left\{3+0.5\bar{\rho},\ L+\epsilon_0+\kappa_\theta+\bar{\rho}\right\}.$$

Then it holds that $\sigma_i \le \bar{\sigma}_1^P = \max\{\sigma_0,\ 3\gamma_2+0.5\bar{\rho}\gamma_2,\ \gamma_2(L+\epsilon_0+\kappa_\theta+\bar{\rho})\}$ for any $i \le T_1$. On the other hand, it follows from the construction of Algorithm 2 that $\sigma_{\min} \le \sigma_i$ for all iterations, and $\gamma_1\sigma_i \le \sigma_{i+1}$ for all unsuccessful iterations. Consequently, we have

$$\frac{\bar{\sigma}_1^P}{\sigma_{\min}} \ge \frac{\sigma_{T_1}}{\sigma_0} = \frac{\sigma_{T_1}}{\sigma_{T_1-1}}\cdot\prod_{j=0}^{T_1-2}\frac{\sigma_{j+1}}{\sigma_j} = \prod_{j=0}^{T_1-2}\frac{\sigma_{j+1}}{\sigma_j} \ge \gamma_1^{T_1-1},$$

where the second equality is due to $\sigma_{T_1} = \sigma_{T_1-1}$ in Algorithm 2, and hence

$$T_1 \le \left(1+\frac{1}{\log\gamma_1}\log\left(\frac{\bar{\sigma}_1^P}{\sigma_{\min}}\right)\right).$$

This completes the proof of Lemma 5. □

Proof of Lemma 6: We have

$$\begin{aligned}
& \mathbf{s}_j^\top \nabla f(\mathbf{y}_l + \mathbf{s}_j) \\
= \;& \mathbf{s}_j^\top [\nabla f(\mathbf{y}_l + \mathbf{s}_j) - \nabla f(\mathbf{y}_l) - \nabla^2 f(\mathbf{y}_l)\mathbf{s}_j] + \mathbf{s}_j^\top [\nabla f(\mathbf{y}_l) + \nabla^2 f(\mathbf{y}_l)\mathbf{s}_j] \\
\leq \;& \|\nabla f(\mathbf{y}_l + \mathbf{s}_j) - \nabla f(\mathbf{y}_l) - \nabla^2 f(\mathbf{y}_l)\mathbf{s}_j\|\|\mathbf{s}_j\| + \mathbf{s}_j^\top [\nabla f(\mathbf{y}_l) + H(\mathbf{y}_l)\mathbf{s}_j + \sigma_j\|\mathbf{s}_j\|\mathbf{s}_j] \\
& + \mathbf{s}_j^\top (\nabla^2 f(\mathbf{y}_l) - H(\mathbf{y}_l))\mathbf{s}_j - \sigma_j\|\mathbf{s}_j\|^3 \\
\overset{(15)}{\leq} \;& \|\nabla f(\mathbf{y}_l + \mathbf{s}_j) - \nabla f(\mathbf{y}_l) - \nabla^2 f(\mathbf{y}_l)\mathbf{s}_j\|\|\mathbf{s}_j\| + (\kappa_\theta - \sigma_j)\|\mathbf{s}_j\|^3 + 2\epsilon_j\|\mathbf{s}_j\|^2 \\
= \;& \|\int_0^1 [\nabla^2 f(\mathbf{y}_l + \tau \cdot \mathbf{s}_j) - \nabla^2 f(\mathbf{y}_l)]\mathbf{s}_j \, d\tau\|\|\mathbf{s}_j\| + (\kappa_\theta - \sigma_j)\|\mathbf{s}_j\|^3 + 2\epsilon_j\|\mathbf{s}_j\|^2 \\
\overset{(5)}{\leq} \;& \left(\frac{\bar{\rho}}{2} + \kappa_\theta - \sigma_j\right)\|\mathbf{s}_j\|^3 + 2\epsilon_j\|\mathbf{s}_j\|^2.
\end{aligned}$$

Next we argue that when σ_i exceeds certain constant, it holds

$$-\frac{\mathbf{s}_j^\top \nabla f(\mathbf{y}_l + \mathbf{s}_j)}{\|\mathbf{s}_j\|^3} \geq \eta,$$

The analysis is conducted according to the value of $\|\mathbf{s}_j\|$ in two cases.

1. When $\|\mathbf{s}_j\| \geq 1$, we have

   $$\mathbf{s}_j^\top \nabla f(\mathbf{y}_l + \mathbf{s}_j) \leq \left(\frac{\bar{\rho}}{2} + \kappa_\theta - \sigma_j + 2\epsilon_j\right)\|\mathbf{s}_j\|^3,$$

   which combined with $\epsilon_j \leq \epsilon_0 \leq 1$ implies that

   $$\sigma_j \geq \frac{\bar{\rho}}{2} + \kappa_\theta + \eta + 2 \Longrightarrow -\frac{\mathbf{s}_j^\top \nabla f(\mathbf{y}_l + \mathbf{s}_j)}{\|\mathbf{s}_j\|^3} \geq \eta.$$

2. When $\|\mathbf{s}_j\| < 1$, similar argument of (27) implies that

   $$\|\mathbf{s}_j\| \geq \frac{\|\nabla f(\mathbf{y}_l)\|}{L + \epsilon_0 + \sigma_i + \kappa_\theta}. \tag{30}$$

   Moreover, recall that

   $$\mathbf{s}_j^\top \nabla f(\mathbf{y}_l + \mathbf{s}_j) \leq \left(\frac{\bar{\rho}}{2} + \kappa_\theta - \sigma_j + \frac{2\epsilon_j}{\|\mathbf{s}_j\|}\right)\|\mathbf{s}_j\|^3.$$

   Combining the above two inequalities yields that

   $$\frac{2\epsilon_j(L + \epsilon_0 + \sigma_i + \kappa_\theta)}{\|\nabla f(\mathbf{y}_l)\|} + \frac{\bar{\rho}}{2} + \kappa_\theta - \sigma_j + \eta \leq 0 \Longrightarrow -\frac{\mathbf{s}_j^\top \nabla f(\mathbf{y}_l + \mathbf{s}_j)}{\|\mathbf{s}_j\|^3} \geq \eta.$$

   Recall in Algorithm 3 that

   $$\epsilon_j = \min\left\{\frac{\|\nabla f(\mathbf{y}_l)\|}{4}, \; \epsilon_0\right\} \leq \frac{\|\nabla f(\mathbf{y}_l)\|}{4},$$

then it suffices to show

$$\frac{L+\epsilon_0+\sigma_j+\kappa_\theta}{2}+\frac{\bar{\rho}}{2}+\kappa_\theta-\sigma_j+\eta\leq 0.$$

That is,

$$\sigma_j\geq L+\epsilon_0+\bar{\rho}+3\kappa_\theta+2\eta\Longrightarrow -\frac{\mathbf{s}_j^\top\nabla f(\mathbf{y}_l+\mathbf{s}_j)}{\|\mathbf{s}_j\|^3}\geq\eta.$$

In summary, we have concluded that

$$\sigma_j\geq\max\left\{\frac{\bar{\rho}}{2}+\kappa_\theta+\eta+2, L+\epsilon_0+\bar{\rho}+3\kappa_\theta+2\eta\right\}\Longrightarrow -\frac{\mathbf{s}_j^\top\nabla f(\mathbf{y}_l+\mathbf{s}_j)}{\|\mathbf{s}_j\|^3}\geq\eta,$$

which further implies that for any unsuccessful iteration $j\notin\mathcal{SC}$, the following inequality holds true,

$$\sigma_j<\max\left\{\frac{\bar{\rho}}{2}+\kappa_\theta+\eta+2,\ L+\epsilon_0+\bar{\rho}+3\kappa_\theta+2\eta\right\}.$$

Therefore, for any successful iteration $j\in\mathcal{SC}$, we have

$$\sigma_{j+1}\leq\sigma_j\leq\gamma_2\cdot\sigma_{j-1}\leq\gamma_2\max\left\{\frac{\bar{\rho}}{2}+\kappa_\theta+\eta+2,\ L+\epsilon_0+\bar{\rho}+3\kappa_\theta+2\eta\right\}.$$

Consequently, for any $0\leq j\leq T_2$, we have

$$\sigma_j\leq\bar{\sigma}_2^P=\max\left\{\bar{\sigma}_1^P,\ \frac{\gamma_2\bar{\rho}}{2}+\gamma_2\kappa_\theta+\gamma_2\eta+2\gamma_2,\ \gamma_2L+\gamma_2\epsilon_0+\gamma_2\bar{\rho}+3\gamma_2\kappa_\theta+2\gamma_2\eta\right\},\quad(31)$$

where $\bar{\sigma}_1^P$ is responsible for an upper bound of σ_0. In addition, it follows from the construction of Algorithm 1 that $\sigma_{\min}\leq\sigma_j$ for all iterations, and $\gamma_1\sigma_j\leq\sigma_{j+1}$ for all unsuccessful iterations. Therefore, we have

$$\frac{\bar{\sigma}_2^P}{\sigma_{\min}}\geq\frac{\sigma_{T_1+T_2}}{\sigma_{T_1}}=\prod_{j\in\mathcal{SC}}\frac{\sigma_{j+1}}{\sigma_j}\cdot\prod_{j\notin\mathcal{SC}}\frac{\sigma_{j+1}}{\sigma_j}\geq\gamma_1^{T_2-|\mathcal{SC}|}\left(\frac{\sigma_{\min}}{\bar{\sigma}_2^P}\right)^{|\mathcal{SC}|},$$

hence

$$|\mathcal{SC}|\leq T_2\leq|\mathcal{SC}|+\frac{(|\mathcal{SC}|+1)}{\log\gamma_1}\log\left(\frac{\bar{\sigma}_2^P}{\sigma_{\min}}\right)\leq\left(1+\frac{2}{\log\gamma_1}\log\left(\frac{\bar{\sigma}_2^P}{\sigma_{\min}}\right)\right)|\mathcal{SC}|.$$

This completes the proof of Lemma 6. □

We present Jiang et al. (2020, Lemma 3.3 and 3.4), which are important to the subsequent analysis.

Lemma 15 *For any $\mathbf{s}\in\mathbb{R}^d$ and $\mathbf{g}\in\mathbb{R}^d$, it holds that*

$$\mathbf{s}^\top\mathbf{g}+\frac{1}{3}\sigma\|\mathbf{s}\|^3\geq-\frac{2}{3\sqrt{\sigma}}\|\mathbf{g}\|^{\frac{3}{2}}.$$

Lemma 16 *Letting $\mathbf{z}_l=\operatorname{argmin}_{\mathbf{z}\in\mathbb{R}^d}\psi_l(\mathbf{z})$, we have $\psi_l(\mathbf{z})-\psi_l(\mathbf{z}_l)\geq(\varsigma_l/12)\|\mathbf{z}-\mathbf{z}_l\|^3$.*

The following lemma is useful to bound the total number of successfully updating $\varsigma > 0$.

Lemma 17 *Suppose in each iteration j of Algorithm 3, we have $\|\nabla^2 f(\mathbf{x}_j) - H(\mathbf{x}_j)\| \leq \epsilon_j$ for any $0 \leq j \leq T_2$. Then we have*

$$\|\nabla f(\mathbf{x}_{j+1})\| \leq \big(0.5\bar{\rho} + 2\kappa_\theta + L + \epsilon_0 + 2\bar{\sigma}_2^P + 2\big)\|\mathbf{s}_j\|^2,$$

where $\kappa_\theta \in (0,1)$ is used in Condition 3.1.

Proof. Note that $\nabla_{\mathbf{s}} m(\mathbf{s}_j; \mathbf{x}_j, \sigma_j) := \nabla f(\mathbf{x}_j) + H(\mathbf{x}_j)\mathbf{s}_j + \sigma_j\|\mathbf{s}_j\| \cdot \mathbf{s}_j$. Then we have

$$\begin{aligned}
& \|\nabla f(\mathbf{x}_{j+1})\| \\
\leq\ & \|\nabla f(\mathbf{x}_{j+1}) - \nabla f(\mathbf{x}_j) - \nabla^2 f(\mathbf{x}_j)\mathbf{s}_j| + \|\nabla^2 f(\mathbf{x}_j) - H(\mathbf{x}_j)\|\|\mathbf{s}_j\| + \sigma_j\|\mathbf{s}_j\|^2 + \|\nabla_{\mathbf{s}} m(\mathbf{s}_j; \mathbf{x}_j, \sigma_j)\| \\
\leq\ & \left\|\int_0^1 (\nabla^2 f(\mathbf{x}_j + \tau\mathbf{s}_j) - \nabla^2 f(\mathbf{x}_j))\mathbf{s}_j d\tau\right\| + 2\epsilon_j\|\mathbf{s}_j\| + \sigma_j\|\mathbf{s}_j\|^2 + \kappa_\theta\|\mathbf{s}_j\|^2 \\
\leq\ & \frac{\bar{\rho}}{2}\|\mathbf{s}_j\|^2 + 2\epsilon_j\|\mathbf{s}_j\| + \bar{\sigma}_2\|\mathbf{s}_j\|^2 + \kappa_\theta \cdot \|\mathbf{s}_j\|^2,
\end{aligned}$$

where the second inequality holds true due to Condition 3.1, and the last inequality follows from Assumption 2 and (31). The subsequent analysis is conducted according to the value of $\|\mathbf{s}_j\|$ in two cases.

1. When $\|\mathbf{s}_j\| \geq 1$, we have

   $$\|\nabla f(\mathbf{x}_{j+1})\| \leq \left(\frac{\bar{\rho}}{2} + 2\epsilon_j + \bar{\sigma}_2^P + \kappa_\theta\right)\|\mathbf{s}_j\|^2,$$

   which combined with $\epsilon_j \leq \epsilon_0 \leq 1$ implies that

   $$\|\nabla f(\mathbf{x}_{j+1})\| \leq \left(\frac{\bar{\rho}}{2} + \bar{\sigma}_2^P + \kappa_\theta + 2\right)\|\mathbf{s}_j\|^2.$$

2. When $\|\mathbf{s}_j\| < 1$, recall in Algorithm 3 that

   $$\epsilon_j = \min\left\{\frac{\|\nabla f(\mathbf{y}_l)\|}{4},\ \epsilon_0\right\} \leq \frac{\|\nabla f(\mathbf{y}_l)\|}{4},$$

   which combined with the first inequality at the beginning of the proof implies that

   $$\begin{aligned}
   \|\nabla f(\mathbf{x}_{j+1})\| &\leq \left(\frac{\bar{\rho}}{2} + \kappa_\theta + \bar{\sigma}_2^P\right)\|\mathbf{s}_j\|^2 + \frac{\|\nabla f(\mathbf{y}_l)\|\|\mathbf{s}_j\|}{2} \\
   &\leq \left(\frac{\bar{\rho}}{2} + \kappa_\theta + \bar{\sigma}_2^P\right)\|\mathbf{s}_j\|^2 + (L + \epsilon_0 + \kappa_\theta + \sigma_j)\|\mathbf{s}_j\|^2 \\
   &\leq \left(\frac{\bar{\rho}}{2} + 2\kappa_\theta + L + \epsilon_0 + 2\bar{\sigma}_2^P\right)\|\mathbf{s}_j\|^2.
   \end{aligned}$$

In summary, we have

$$\|\nabla f(\mathbf{x}_{j+1})\| \leq \left(\frac{\bar{\rho}}{2} + 2\kappa_\theta + L + \epsilon_0 + 2\bar{\sigma}_2^P + 2\right)\|\mathbf{s}_j\|^2.$$

□

Proof of Lemma 7: We are now ready to provide an upper bound of T_3. When $l = 0$, it trivially holds true that $\psi_l(\mathbf{z}_l) \geq (1/6)l(l+1)(l+2)f(\bar{\mathbf{x}}_l)$ since $\psi_0(\mathbf{z}) = f(\bar{\mathbf{x}}_0)$. It suffices to establish the general case when $\varsigma_l \geq 8\eta^{-2}(0.5\bar{\rho} + 2\kappa_\theta + L + 2\bar{\sigma}_2^P + 1)^3$ by mathematical induction. Without loss of generality, we assume (17) holds true for some $l - 1 \geq 1$. Then, it follows from Lemma 16, and the construction of $\psi_l(\mathbf{z})$ that

$$\psi_{l-1}(\mathbf{z}) \geq \psi_{l-1}(\mathbf{z}_{l-1}) + \frac{1}{12}\varsigma_{l-1}\|\mathbf{z} - \mathbf{z}_{l-1}\|^3 \geq \frac{(l-1)l(l+1)}{6}f(\bar{\mathbf{x}}_{l-1}) + \frac{1}{12}\varsigma_{l-1}\|\mathbf{z} - \mathbf{z}_{l-1}\|^3.$$

As a result, we have

$$\begin{aligned}
& \psi_l(\mathbf{z}_l) \\
= & \min_{\mathbf{z}\in\mathbb{R}^d} \left\{\psi_{l-1}(\mathbf{z}) + \frac{l(l+1)}{2}[f(\bar{\mathbf{x}}_l) + (\mathbf{z} - \bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l)] + \frac{1}{6}(\varsigma_l - \varsigma_{l-1})\|\mathbf{z} - \bar{\mathbf{x}}_0\|^3\right\} \\
\geq & \min_{\mathbf{z}\in\mathbb{R}^d} \left\{\frac{(l-1)l(l+1)}{6}f(\bar{\mathbf{x}}_{l-1}) + \frac{\varsigma_{l-1}}{12}\|\mathbf{z} - \mathbf{z}_{l-1}\|^3 + \frac{l(l+1)}{2}\left[f(\bar{\mathbf{x}}_l) + (\mathbf{z} - \bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l)\right]\right\} \\
\geq & \min_{\mathbf{z}\in\mathbb{R}^d} \left\{\frac{(l-1)l(l+1)}{6}[f(\bar{\mathbf{x}}_l) + (\bar{\mathbf{x}}_{l-1} - \bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l)] + \frac{\varsigma_{l-1}}{12}\|\mathbf{z} - \mathbf{z}_{l-1}\|^3 + \frac{l(l+1)}{2}[f(\bar{\mathbf{x}}_l) + (\mathbf{z} - \bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l)]\right\} \\
= & \frac{l(l+1)(l+2)}{6}f(\bar{\mathbf{x}}_l) + \min_{\mathbf{z}\in\mathbb{R}^d}\left\{\frac{(l-1)l(l+1)}{6}(\bar{\mathbf{x}}_{l-1} - \bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l) + \frac{\varsigma_{l-1}}{12}\|\mathbf{z} - \mathbf{z}_{l-1}\|^3 + \frac{l(l+1)}{2}(\mathbf{z} - \bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l)\right\},
\end{aligned}$$

where the first inequality follows from $\varsigma_l \geq \varsigma_{l-1}$. By the construction of $\mathbf{y}_{l-1}$, we have

$$\begin{aligned}
\frac{(l-1)l(l+1)}{6}\bar{\mathbf{x}}_{l-1} &= \frac{l(l+1)(l+2)}{6}\cdot\frac{l-1}{l+2}\bar{\mathbf{x}}_{l-1} \\
&= \frac{l(l+1)(l+2)}{6}\left(\mathbf{y}_{l-1} - \frac{3}{l+2}\mathbf{z}_{l-1}\right) \\
&= \frac{l(l+1)(l+2)}{6}\mathbf{y}_{l-1} - \frac{l(l+1)}{2}\mathbf{z}_{l-1}.
\end{aligned}$$

Combining the above two formulas yields

$$\begin{aligned}
\psi_l(\mathbf{z}_l) \geq & \frac{l(l+1)(l+2)}{6}f(\bar{\mathbf{x}}_l) + \min_{\mathbf{z}\in\mathbb{R}^d}\left\{\frac{l(l+1)(l+2)}{6}(\mathbf{y}_{l-1} - \bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l)\right. \\
& \left. + \frac{\varsigma_{l-1}}{12}\|\mathbf{z} - \mathbf{z}_{l-1}\|^3 + \frac{l(l+1)}{2}(\mathbf{z} - \mathbf{z}_{l-1})^\top \nabla f(\bar{\mathbf{x}}_l)\right\}.
\end{aligned}$$

By the criterion of successful iteration in Algorithm 3 and Lemma 17, we have

$$(\mathbf{y}_{l-1} - \bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l) = -\mathbf{s}_j^\top \nabla f(\bar{\mathbf{x}}_l) \geq \eta\|\mathbf{s}_j\|^3 \geq \eta\left(\frac{\|\nabla f(\bar{\mathbf{x}}_l)\|}{0.5\bar{\rho} + 2\kappa_\theta + L + \epsilon_0 + 2\bar{\sigma}_2^P + 2}\right)^{\frac{3}{2}},$$

where the l-th successful iteration count refers to the j-th iteration count. Hence, it suffices to establish

$$\frac{l(l+1)(l+2)\eta}{6}\left(\frac{\|\nabla f(\bar{\mathbf{x}}_l)\|}{\frac{\bar{\rho}}{2} + 2\kappa_\theta + L + \epsilon_0 + 2\bar{\sigma}_2^P + 2}\right)^{\frac{3}{2}} + \frac{\varsigma_{l-1}}{12}\|\mathbf{z} - \mathbf{z}_{l-1}\|^3 + \frac{l(l+1)}{2}(\mathbf{z} - \mathbf{z}_{l-1})^\top \nabla f(\bar{\mathbf{x}}_l) \geq 0.$$

Using Lemma 15 and setting $\mathbf{g} = 0.5l(l+1)\nabla f(\bar{\mathbf{x}}_l)$, $\mathbf{s} = \mathbf{z} - \mathbf{z}_l$, and $\sigma = \varsigma_{l-1}/4$, the above is implied by

$$\frac{l(l+1)(l+2)\eta}{6}\left(\frac{1}{0.5\bar{\rho}+2\kappa_\theta+L+\epsilon_0+2\bar{\sigma}_2^P+2}\right)^{\frac{3}{2}} \geq \frac{4}{3\sqrt{\varsigma_{l-1}}}\left(\frac{l(l+1)}{2}\right)^{\frac{3}{2}}. \tag{32}$$

Therefore, the conclusion follows if

$$\varsigma_{l-1} \geq 8\eta^{-2}(0.5\bar{\rho}+2\kappa_\theta+L+\epsilon_0+2\bar{\sigma}_2^P+2)^3.$$

This completes the proof.

Appendix B. Proofs in Section 5

We prove Lemma 10 and Lemma 11, which describe the relation between the total iteration numbers in Algorithm 2 and the amount of successful iterations $|\mathcal{SC}|$ in Algorithm 3.

Proof of Lemma 10: According to Condition 3.1, it holds that

$$\begin{array}{rcl} \kappa_\theta\|\nabla f(\mathbf{x}_i)\| & \geq & \|\nabla m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i))\| \;=\; \|\nabla f(\mathbf{x}_i)+H(\mathbf{x}_i)\mathbf{s}_i+\sigma_i\|\mathbf{s}_i\|\cdot\mathbf{s}_i\| \\ & \overset{(19)}{\geq} & \|\nabla f(\mathbf{x}_i)\|-(L+\epsilon_0)\|\mathbf{s}_i\|-\sigma_i\|\mathbf{s}_i\|^2, \end{array}$$

which implies that

$$\sigma_i\|\mathbf{s}_i\|^2+(L+\epsilon_0)\|\mathbf{s}_i\|-(1-\kappa_\theta)\sqrt{\epsilon} \geq 0,$$

and hence

$$\|\mathbf{s}_i\| \geq \frac{-(L+\epsilon_0)+\sqrt{(L+\epsilon_0)^2+4\sigma_i\sqrt{\epsilon}(1-\kappa_\theta)}}{2\sigma_i}. \tag{33}$$

Moreover, we have that

$$\begin{array}{rcl} f(\mathbf{x}_i+\mathbf{s}_i) & = & f(\mathbf{x}_i)+\mathbf{s}_i^\top\nabla f(\mathbf{x}_i)+\displaystyle\int_0^1[\nabla f(\mathbf{x}_i+\tau\mathbf{s}_i)-\nabla f(\mathbf{x}_i)]\,\mathbf{s}_i\,d\tau \\ & \overset{(4)}{\leq} & f(\mathbf{x}_i)+\mathbf{s}_i^\top\nabla f(\mathbf{x}_i)+\dfrac{L}{2}\|\mathbf{s}_i\|^2 \\ & = & m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i))+\dfrac{L}{2}\|\mathbf{s}_i\|^2-\dfrac{1}{2}\mathbf{s}_i^\top H(\mathbf{x}_i)\mathbf{s}_i-\dfrac{\sigma_i}{3}\|\mathbf{s}_i\|^3 \\ & \overset{(19)}{\leq} & m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i))+\left(\dfrac{L}{\|\mathbf{s}_i\|}-\dfrac{\sigma_i}{3}\right)\|\mathbf{s}_i\|^3. \end{array} \tag{34}$$

Combining (33) and (34) yields the following relation

$$\frac{2\sigma_i L}{-(L+\epsilon_0)+\sqrt{(L+\epsilon_0)^2+4\sigma_i\sqrt{\epsilon}(1-\kappa_\theta)}}-\frac{\sigma_i}{3} \leq 0 \Longrightarrow f(\mathbf{x}_i+\mathbf{s}_i) \leq m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i)).$$

Note that the left hand side inequality is equivalent to

$$\frac{(L+\epsilon_0)+\sqrt{(L+\epsilon_0)^2+4\sigma_i\sqrt{\epsilon}(1-\kappa_\theta)}}{2\sqrt{\epsilon}(1-\kappa_\theta)}-\frac{\sigma_i}{3L} \leq 0,$$

which is implied by $\sigma_i \geq \frac{3L(4L+\epsilon_0)}{(1-\kappa_\theta)\sqrt{\epsilon}}$. In summary, we have concluded that

$$\sigma_i \geq \frac{3L(4L+\epsilon_0)}{(1-\kappa_\theta)\sqrt{\epsilon}} \Longrightarrow f(\mathbf{x}_i+\mathbf{s}_i) \leq m(\mathbf{s}_i;\mathbf{x}_i,\sigma_i)).$$

The remaining proof is similar to the argument below (29) in Lemma 5.

Proof of Lemma 11: We have

$$\begin{aligned}
\mathbf{s}_j^\top \nabla f(\mathbf{y}_l+\mathbf{s}_j) &= \mathbf{s}_j^\top\left[\nabla f(\mathbf{y}_l+\mathbf{s}_j)-\nabla f(\mathbf{y}_l)\right]+\mathbf{s}_j^\top\left[\nabla f(\mathbf{y}_l)+H(\mathbf{y}_l)\mathbf{s}_j\right]-\mathbf{s}_j^\top H(\mathbf{y}_l)\mathbf{s}_j\\
&\le \|\nabla f(\mathbf{y}_l+\mathbf{s}_j)-\nabla f(\mathbf{y}_l)\|\|\mathbf{s}_j\|+\mathbf{s}_j^\top[\nabla m(\mathbf{y}_l,\mathbf{s}_j,\sigma_j)-\sigma_j\|\mathbf{s}_j\|\mathbf{s}_j]+\|H(\mathbf{y}_l)\|\|\mathbf{s}_j\|^2\\
&\overset{(4)(15)(19)}{\le} L\|\mathbf{s}_j\|^2+\kappa_\theta\|\mathbf{s}_j\|^3-\sigma_j\|\mathbf{s}_j\|^3+(L+\epsilon_0)\|\mathbf{s}_j\|^2\\
&= \left(\frac{2L+\epsilon_0}{\|\mathbf{s}_j\|}+\kappa_\theta-\sigma_j\right)\|\mathbf{s}_j\|^3.
\end{aligned}$$

A similar argument of (33) implies that

$$\|\mathbf{s}_j\|\ge\frac{-(L+\epsilon_0)+\sqrt{(L+\epsilon_0)^2+4\sigma_j\sqrt{\epsilon}(1-\kappa_\theta)}}{2\sigma_j}.$$

Now combining the two inequalities above yields the following relation.

$$\frac{L+\epsilon_0+\sqrt{(L+\epsilon_0)^2+4\sqrt{\epsilon}\sigma_j(1-\kappa_\theta)}}{2\sqrt{\epsilon}(1-\kappa_\theta)}-\frac{\sigma_j-\kappa_\theta-\eta}{2L+\epsilon_0}\le 0\Longrightarrow -\frac{\mathbf{s}_j^\top\nabla f(\mathbf{y}_l+\mathbf{s}_j)}{\|\mathbf{s}_j\|^3}\ge\eta.$$

A straight forward calculation shows that the inequality on the left hand side is implied by

$$\sigma_j\ge\frac{(3L+2\epsilon_0)(2L+\epsilon_0)+2\sqrt{\epsilon}(1-\kappa_\theta)(\kappa_\theta+\eta)+(2L+\epsilon_0)\sqrt{(3L+2\epsilon_0)^2+\sqrt{\epsilon}(1-\kappa_\theta)(\kappa_\theta+\eta)}}{2\sqrt{\epsilon}(1-\kappa_\theta)}.$$

The remaining proof is similar to the argument in Lemma 6.

Lemma 18 *Suppose in each iteration j of Algorithm 3, we have $\|\nabla f(\mathbf{x}_j)\|^2>\epsilon$ for any $0\le j\le T_2$. Then we have*

$$\|\nabla f(\mathbf{x}_{j+1})\|\le\left((2L+\epsilon_0)\cdot\frac{(L+\epsilon_0)+\sqrt{(L+\epsilon_0)^2+4\bar{\sigma}_2^W\sqrt{\epsilon}(1-\kappa_\theta)}}{2\sqrt{\epsilon}(1-\kappa_\theta)}+\bar{\sigma}_2^W+\kappa_\theta\right)\|\mathbf{s}_j\|^2$$

where $\kappa_\theta\in(0,1)$ is used in Condition 3.1.

Proof. Recalling $\nabla_{\mathbf{s}}m(\mathbf{s}_j;\mathbf{x}_j,\sigma_j)=\nabla f(\mathbf{x}_j)+H(\mathbf{x}_j)\mathbf{s}_j+\sigma_j\|\mathbf{s}_j\|\cdot\mathbf{s}_j$, we have

$$\begin{aligned}
\|\nabla f(\mathbf{x}_{j+1})\| &\le \|\nabla f(\mathbf{x}_j+\mathbf{s}_j)-\nabla_{\mathbf{s}}m(\mathbf{s}_j;\mathbf{x}_j,\sigma_j)\|+\|\nabla_{\mathbf{s}}m(\mathbf{s}_j;\mathbf{x}_j,\sigma_j)\|\\
&\overset{(15)}{\le} \|\nabla f(\mathbf{x}_j+\mathbf{s}_j)-\nabla_{\mathbf{s}}m(\mathbf{s}_j;\mathbf{x}_j,\sigma_j)\|+\kappa_\theta\|\mathbf{s}_j\|^2\\
&\le \|\nabla f(\mathbf{x}_j+\mathbf{s}_j)-\nabla f(\mathbf{x}_j)\|+\|H(\mathbf{x}_j)\|\|\mathbf{s}_j\|+\sigma_j\|\mathbf{s}_j\|^2+\kappa_\theta\|\mathbf{s}_j\|^2\\
&\overset{(4)(19)}{\le} L\|\mathbf{s}_j\|+(L+\epsilon_0)\|\mathbf{s}_j\|+\sigma_j\|\mathbf{s}_j\|^2+\kappa_\theta\|\mathbf{s}_j\|^2\\
&= \left(\frac{2L+\epsilon_0}{\|\mathbf{s}_j\|}+\bar{\sigma}_2^W+\kappa_\theta\right)\|\mathbf{s}_j\|^2.
\end{aligned}$$

A similar argument of (33) implies that

$$\|\mathbf{s}_j\|\ge\frac{-(L+\epsilon_0)+\sqrt{(L+\epsilon_0)^2+4\sigma_j\sqrt{\epsilon}(1-\kappa_\theta)}}{2\sigma_j}.$$

Therefore, we conclude that

$$\|\nabla f(\mathbf{x}_{j+1})\| \leq \left((2L+\epsilon_0) \cdot \frac{(L+\epsilon_0) + \sqrt{(L+\epsilon_0)^2 + 4\bar{\sigma}_2^W \sqrt{\epsilon}(1-\kappa_\theta)}}{2\sqrt{\epsilon}(1-\kappa_\theta)} + \bar{\sigma}_2^W + \kappa_\theta \right) \|\mathbf{s}_j\|^2.$$

□

We are now ready to provide an upper bound of T_3.

Proof of Lemma 12: The proof is almost the same as that of Lemma 7 by mathematical induction. The only difference is the estimation of

$$\begin{aligned} & (\mathbf{y}_{l-1} - \bar{\mathbf{x}}_l)^\top \nabla f(\bar{\mathbf{x}}_l) \\ \geq \quad & \eta \left(\frac{2\sqrt{\epsilon}(1-\kappa_\theta)}{(2L+\epsilon_0)\left((L+\epsilon_0) + \sqrt{(L+\epsilon_0)^2 + 4\bar{\sigma}_2^W \sqrt{\epsilon}(1-\kappa_\theta)}\right) + 2\sqrt{\epsilon}(1-\kappa_\theta)\left(\bar{\sigma}_2^W + \kappa_\theta\right)} \right)^{\frac{3}{2}} \|\nabla f(\bar{\mathbf{x}}_l)\|^{\frac{3}{2}}, \end{aligned}$$

which is due to Lemma 18. By adapting the proof of Lemma 7 with such estimation, we achieve the desired result.