Abstract

In this paper we propose a unified two-phase scheme to accelerate any high-order regularized tensor approximation approach on the smooth part of a composite convex optimization model. The proposed scheme has the advantage of not needing to assume any prior knowledge of the Lipschitz constants for the gradient, the Hessian and/or high-order derivatives. This is achieved by tuning the parameters used in the algorithm adaptively in its process of progression, which can be viewed as a relaxation over the existing algorithms in the literature. Under the assumption that the sub-problems can be solved approximately, we establish the overall iteration complexity bounds for three specific algorithms to obtain an $\epsilon$-optimal solution. In general, we show that the adaptive high-order method has an iteration bound of $O\left(1/\epsilon^{1/(p+1)}\right)$ if the first $p$-th order derivative information is used in the approximation, which has the same iteration complexity as in [39] where the Lipschitz constants are assumed to be known and subproblems are assumed to be solved exactly. Thus, our results answer an open problem raised by Nesterov in [39] on adaptive strategies for high-order accelerated methods. Specifically, we show that the gradient method achieves an iteration complexity in the order of $O\left(1/\epsilon^{1/2}\right)$, which is known to be best possible (cf. [38]), while the adaptive cubic regularization methods with the exact/inexact Hessian matrix both achieve an iteration complexity in the order of $O\left(1/\epsilon^{1/3}\right)$, which matches that of the original accelerated cubic regularization method presented in [36] assuming the availability of the exact Hessian information and the Lipschitz constants, and the global solution of the sub-problems. Our numerical experiment results show a clear effect of acceleration displayed in the adaptive Newton’s method with cubic regularization on a set of regularized logistic regression instances.

Keywords: convex optimization; tensor method; acceleration; adaptive method; iteration complexity.

Mathematics Subject Classification: 90C06, 90C60, 90C53.
1 Introduction

1.1 Motivations

In this paper, we consider the following generic composite convex optimization model:

\[ F^* := \min_{x \in \mathbb{R}^d} F(x) = f(x) + r(x), \]  

where \( f : \mathbb{R}^d \to \mathbb{R} \) is convex and smooth, \( r : \mathbb{R}^d \to \mathbb{R} \) is convex but possibly nonsmooth with simple proximal mapping, and \( F^* > -\infty \). During the past decades, various classes of optimization algorithms for solving (1) (especially when \( r(x) = 0 \) and \( F(x) \) becomes smooth) have been developed and carefully analyzed; see [30, 41, 38] for relevant information and references therein. Theoretical niceties of the proposed solution methods aside, there has been a practical concern regarding the implementation, as many methods assume that some problem parameters such as the first and the second order Lipschitz constants are available, which may be hard to estimate in practice. It will be ideal to come up with optimization algorithms which automatically estimate such parametric values, making the algorithms easy-implementable while maintaining superior theoretical iteration bounds intact. In this case, we are demanding an algorithm to be less dependent on the knowledge of the problem structure at hand, therefore less prone to failures due to misinformation of such values. In this context, schemes that adaptively adjust the parameters used in the algorithms are often desirable, and are likely leading to better numerical performances. For instance, researchers tend to train their deep learning models with adaptive gradient method (see e.g. AdaGrad in [18]) due to its robustness and effectiveness (cf. [24]). In fact, Adam [25] and RMSProp [47] are recognized as the default solution methods in the deep learning setting. Among the category of second order methods, Cartis et al. [11, 12] proposed and analyzed an adaptive cubic regularized Newton’s method, which soon became very popular due to its numerical efficiency. In a very recent working paper [36], Nesterov proposed two implementable high-order methods where he also commented that an unsolved issue in his approach was a dynamic adjustment scheme for the Lipschitz constant for the highest derivative to achieve practical efficiency.

Another fundamental issue in optimization (as well as in machine learning) is to understand how the classical algorithms (including the first-order, second-order and high-order methods) can be accelerated. Nesterov [35] put forward the very first accelerated (optimal in its iteration counts) gradient-based algorithm for smooth convex optimization. Beck and Teboulle [4] successfully extended Nesterov’s approach to accomodate the problem in the form of (1). Recently, a number of adaptively accelerated gradient methods have been proposed; see [18, 37, 29, 32, 42, 9]. Among those, the algorithms in [42, 9] are fully problem-parameter-free. Comparing to their first-order counterpart, investigations on the second-order or high-order methods are relatively scarce, as acceleration with the high-order information is much more involved. To the best of our knowledge, [36, 34, 19] are the only papers that are concerned with accelerating the second-order methods, and [36] is the only one that studies acceleration for high (higher than 2) order tensor approximation method. However, these algorithms do require the knowledge of some problem-specific parametric (Lipschitz) constants.

Overall, algorithms exhibiting both traits of acceleration and adaptation have been largely missing in the literature. As a matter of fact, we are unaware of any prior accelerated second-order methods (or any high-order methods) that are fully independent of the problem constants while maintaining superior theoretical iteration bounds. For instance, the adaptive cubic regularized Newton’s method
13] is Hessian-free and problem-parameter-free, and allows subproblem to be solved inexactly, but it merely achieves an iteration bound of $O\left(1/\epsilon^{1/2}\right)$ without acceleration. Thus, a natural question raises: Can we develop an implementable accelerated second-order method with an iteration complexity lower than $O\left(1/\epsilon^{1/2}\right)$? One goal of this paper is to present an affirmative answer to this question. It turns out the resulting accelerated adaptive cubic regularization algorithm displays an excellent numerical performance in solving a variety of large-scale machine learning models in our experiments. Furthermore, the analysis is extended to accelerating any high-order tensor approximation approach while allowing the objective to be composite: a part of the objective function need not be smooth at all.

1.2 Related Work

Nesterov’s seminal work [35] triggered a burst of research on accelerating first-order methods. There have been a good deal of recent efforts trying to understand its nature from other perspectives [3, 7, 46, 48, 49], or modify it to account for more general settings [4, 16, 26, 17, 44, 28]. Parallel to this, the adaptive gradient methods with the optimal convergence rate have been proposed [18, 37, 29, 32], and widely used in training the deep neural networks [25, 47]. However, all of these algorithms are not fully problem-parameter-free. Specifically, Duchi et al. [18] needs to tune the step-size and the regularization parameter; Lin and Xiao [29] and Nesterov [37] require a lower bound on the Lipschitz constant $L$ for the gradient; and Monteiro and Svaiter [32] need an upper bound of $L - \mu$, where $\mu$ is a strong convexity parameter. The problem-parameter-free accelerated first-order algorithms were recently proposed in [42, 9].

In terms of the second-order methods (in particular Newton’s method), the literature regarding acceleration is quite limited. To the best of our knowledge, Nesterov [36] is the first along this direction, where the overall iteration complexity for convex optimization was improved from $O \left(1/\epsilon^{1/2}\right)$ to $O \left(1/\epsilon^{1/3}\right)$ for the cubic regularized Newton’s method [40]. After that, Monteiro and Svaiter [34] managed to accelerate the Newton proximal extragradient method with an improved iteration complexity of $O \left(1/\epsilon^{2/7}\right)$. Moreover, this approach allows a larger stepsize and can even accommodate a non-smooth objective function. Very recently, Shamir and Shiff [45] proved that $O \left(1/\epsilon^{2/7}\right)$ is actually a lower bound for the oracle complexity of the second-order methods for convex smooth optimization, which implies that the accelerated Newton proximal extragradient method is an optimal second-order method. However, viewed from an implementation perspective, the acceleration second-order scheme in [36, 34] are not easy to apply in practice. Indeed, Nesterov’s method assumes that all the parameters, including the Lipschitz constant for the Hessian, are known, and the sub-problems with cubic regularization are solved to global optimality; Monteiro and Svaiter’s method also assumes the knowledge of the Lipschitz constant of the Hessian. To alleviate this, Cartis et al. incorporated an adaptive strategy into Nesterov’s approach [36], and further relaxed the criterion for solving each sub-problem while maintaining the convergence properties for both convex [13] and non-convex [11, 12] cases. However, as mentioned earlier, the iteration complexity established in [13] for convex optimization is merely $O \left(1/\epsilon^{1/2}\right)$. Furthermore, in [14] the same authors also developed a way to construct an approximation for the Hessian, which significantly reduces the per-iteration cost. There are other recent works on approximate cubic regularization for Newton’s method. For instance, Carmon and Duchi [10] and Agarwal et al. [1] proposed some variants, where the sub-problem is approximately solved without resorting to Hessian matrix; Xu et al. [50] proposed appropriate uniform and non-uniform sub-sampling strategies to construct Hessian approximations in the cubic regularization for Newton’s method. In a recent work [19], Ghadimi et al.
generalized accelerated Newton’s method with cubic regularization under inexact second-order information. However, the complexity bound is theoretically worse than that of its exact counterparts, and only as good as that of the optimal first-order method. Very recently, Nesterov [39] proposed a high-order generalization of accelerated cubic regularized Newton’s method with an improved iteration bound of $O(1/\epsilon^{(p+1)/p})$ when up to $p$-th order information is utilized. The high-order methods for nonconvex optimization were also studied in [6, 15, 31].

It is worth noting that the standard cubic regularized Newton’s method and its variants are tailored for smooth unconstrained convex optimization, and thus is unsuited to solve (1) directly. In the literature, there are second-order methods which are efficient for solving (1), and they are referred to as proximal (quasi-)Newton methods. The global convergence and the local superlinear rate of convergence of those methods have been shown in [27] and more recently in [8]. The global sublinear rate of $O(1/\epsilon)$ for proximal quasi-Newton methods is established in [43] and was later accelerated to $O(1/\epsilon^{1/2})$ in [20]. [5] proposed a highly efficient evaluation of the proximal mapping within the quasi-Newton framework. Very recently, Grapiglia and Nesterov [21] extended accelerated regularized Newton’s Methods to solve problem (1), where $f$ is twice differentiable with a Hölderian continuous Hessian, and they showed that the iteration bound depends on the Hölderian parameter.

1.3 Contributions

The contributions of this paper can be summarized as follows. We present a unified adaptive accelerating scheme that can be specialized to several optimization algorithms including gradient method, cubic regularized Newton’s method with exact/inexact Hessian and high-order method. This can be considered complementary to the current stream of research in four aspects. First, all the accelerated algorithms developed in this paper are problem-parameter-free due to the adopted fully adaptive strategies, while all the other accelerated second-order (high-order) methods in the literature require the knowledge of some problem parameters. Our results indeed answer an open problem raised by Nesterov in [39] regarding a dynamic adjustment scheme to estimate the Lipschitz constant in a high-order method to achieve practical efficiency. Note that the accelerated first-order methods proposed in [42, 9] shared the same merit, albeit their analysis is quite different. Second, we observe that the research efforts on accelerated algorithms have been rather unequally spread between the first-order and second-order (and higher-order) methods, with the former receiving a lot more attention. Our results on the adaptive and accelerated cubic regularization for Newton’s method and high-order method contribute as one step towards balancing the studies on those methods. Third, we only require an approximative solution satisfying (9) for the subproblem considered in our framework. In the context of cubic regularized Newton’s method, our approximativeness measure does not include the usual condition in the form of (10), thus is weaker than the one used in [11] and similar to the one in [6] for nonconvex optimization. This relaxation opens up possibilities for other approximation solution methods to solve the subproblems. For instance, Carmon and Duchi [10] proposed to use the gradient descent method, and they proved that it works well even when the cubic regularized subproblem is nonconvex. Moreover, the merit function in our case is strongly convex, and thus the gradient descent subroutine is expected to have a fast (linear) convergence. Lastly, in this case the cubic regularized Newton’s method is extended to minimizing the sum of two convex functions: one of which maybe nonsmooth. Prior to this work, such methods are mostly proposed for unconstrained smooth convex optimization. As we were finalising this manuscript, we noticed a very recent working paper by Grapiglia and Nesterov [21] which also solves the problem.
in the form of (1) with the focus on exploring the Hölder continuity of the Hessian, and study how this affects the iteration bound of the algorithm. In comparison, our framework focuses on the high-order approximation, adaptive strategy and inexact solutions of the subproblems.

In terms of the convergence rates of our algorithms, for the gradient descent method, our adaptive algorithm achieves a convergence rate of $O\left(\frac{1}{\epsilon^{1/2}}\right)$ (Theorem 4.1) which matches the optimal rate for the first order methods [38]. For the cubic regularized Newton’s method we show that a global convergence rate of $O\left(\frac{1}{\epsilon^{1/3}}\right)$ holds (Theorem 4.2) without assuming any knowledge of the problem parameters. We further prove that, even without the exact Hessian information, the same $O\left(\frac{1}{\epsilon^{1/3}}\right)$ rate of convergence (Theorem 4.3) is still achievable for the cubic regularized approximative Newton’s method. When our adaptive scheme reduces to the high-order method, the global rate of $O\left(\frac{1}{\epsilon^{1/(p+1)}}\right)$ is guaranteed by utilizing the up to $p$-th order information, which achieves the same iteration bound as in Nesterov [39].

1.4 Notations and Organization

Notations: 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 $\ell_2$ norm and the matrix spectral norm, respectively. $\nabla f(\mathbf{x})$, $\nabla^2 f(\mathbf{x})$ and $\nabla^d f(\mathbf{x})$ are respectively the gradient, the Hessian and $p$-th order derivative tensor of $f$ at $\mathbf{x}$. We denote

$$\nabla^d f(\mathbf{x})[x^1, \ldots, x^d] := \sum_{i_1, \ldots, i_d=1}^{n} \nabla^d f(\mathbf{x})_{i_1, \ldots, i_d} x^1_{i_1} \cdots x^d_{i_d},$$

and $I$ denotes the identity matrix. For two symmetric matrices $A$ and $B$, $A \succeq B$ indicates that $A - B$ is symmetric positive semidefinite. The $\log(x)$ denotes the natural logarithm of $x$ for $x > 0$.

Organization: The rest of the paper is organized as follows. In Section 2, we introduce some preliminaries and the assumptions used throughout this paper. In Section 3, we propose our general framework to adaptively accelerate various optimization algorithms, and present the main theoretical results on the iteration complexity. Section 4 is devoted to specializations of our framework to first-order methods, second-order methods, and high-order methods. In Section 5, we present some preliminary numerical results on solving $\ell_2$-regularized and $\ell_1$-regularized logistic regression problems, where acceleration of the method based on the adaptive cubic regularization for Newton’s method is clearly observed. The details of all the proofs can be found in the appendix.

2 Preliminaries

Throughout this paper, we make the following assumptions for problem (1).

Assumption 2.1 $F$ is a proper, closed and convex function in the domain

$$\text{dom}(F) := \left\{ \mathbf{x} \in \mathbb{R}^d \mid F(\mathbf{x}) < +\infty \right\},$$

and the optimal set of problem (1) is nonempty.
Assumption 2.2 The function $f$ is $p$-th continuously differentiable and $\nabla^j f$ is Lipschitz continuous with $L_j > 0$ for $p - 1 \leq j \leq p$, i.e.,

$$\|\nabla^j f(x) - \nabla^j f(y)\| \leq L_j \|x - y\|, \quad \forall x, y \in \text{dom}(F).$$  

(2)

We remark that Assumption 2.2 is standard in the convergence analysis of $p$-th order optimization methods for minimizing smooth functions ([6, 39]). We consider the following $p$-th order approximation of $f(y)$ at point $x$:

$$\tilde{f}_p(y; x) = f(x) + (y - x)^\top \nabla f(x) + \frac{1}{2} (y - x)^\top \nabla^2 f(x) (y - x) + \sum_{j=3}^{p} \frac{1}{j!} \nabla^j f(x) \left[ y - x, \ldots, y - x \right].$$

Under Assumptions 2.1-2.2, the following two inequalities follow from residual analysis for the Taylor expansion (see also [39]):

$$|f(y) - \tilde{f}_p(y; x)| \leq \frac{L_p \|y - x\|^{p+1}}{(p + 1)!},$$  

(3)

and

$$\|\nabla f(y) - \nabla \tilde{f}_p(y; x)\| \leq \frac{L_p \|y - x\|^p}{p!}.$$  

(4)

Based on $\tilde{f}_p(y; x)$, we consider other approximations of $f(y)$. We call function $m(y; x)$ an effective approximation of the smooth function $f(y)$ at point $x$ if the following properties hold.

Definition 2.1 We call $m(y; x)$ to be an effective approximation of $f(y)$ at a given point $x \in \text{dom}(F)$ if it satisfies the following three properties:

(i) For any $y \in \text{dom}(F)$, it holds that

$$|f(y) - m(y; x)| \leq \bar{\kappa}_p \|y - x\|^p + \kappa_p \|y - x\|^{p+1}$$  

(5)

for some constants $\bar{\kappa}_p$ and $\kappa_p$.

(ii) For any $\bar{x} \approx \arg\min_{y \in \mathbb{R}^d} m(y; x, \sigma)$, it holds that

$$|f(\bar{x}) - m(\bar{x}; x)| \leq \beta_p \|\bar{x} - x\|^{p+1},$$  

(6)

$$\|\nabla f(\bar{x}) - \nabla m(\bar{x}; x)\| \leq \rho_p \|\bar{x} - x\|^p;$$  

(7)

or the above two inequalities hold for any $\|\bar{x} - x\| \geq h$ when $m(\bullet; x)$ is additionally dependent on some positive number $h$, where all the parameters are constants.

(iii) $m(y; x)$ is convex in $y$.

The specific choices of $m(y; x)$ and the corresponding values of $\bar{\kappa}_p, \kappa_p, \beta_p, \rho_p$ are provided in Table 1; more details will be provided in Section 4.

We end this section by specifying the definitions of $\varepsilon$-optimality and proximal mapping which are frequently used in this paper.
| Val. of $p$ | Exact/Inexact Derivative | $\overline{m}(y; x)$ | $\kappa_p$ | $\kappa_p$ | $\beta_p$ | $\rho_p$ |
|---|---|---|---|---|---|---|
| 1 | Exact | $f_1(y; x)$ | 0 | $\frac{L_1}{p}$ | $\frac{L_1}{p}$ | $L_1$ |
| 2 | Exact | $f_2(y; x)$ | 0 | $\frac{L_2}{p}$ | $\frac{L_2}{p}$ | $L_2$ |
| 2 | Inexact Hessian | $f_1(y; x) + \frac{(y - x)^\top H(x)(y - x)}{1}$ | $L_1$ | $\frac{L_2}{p}$ | $\frac{L_2}{p}$ | $L_2$ |
| $p$ | Exact | $f_p(y; x)$ | 0 | $\frac{L_p}{(p+1)!}$ | $\frac{L_p}{(p+1)!}$ | $L_p$ |

Table 1: Specific choices of $\overline{m}(y; x)$

**Definition 2.2 (ε-optimality)** Given $\varepsilon \in (0, 1)$, $x \in \mathbb{R}^d$ is said to be $\varepsilon$-optimal to problem (1) if

$$F(x) - F(x^*) \leq \varepsilon,$$

where $x^* \in \mathbb{R}^d$ is an optimal solution to problem (1).

**Definition 2.3 (proximal mapping)** The proximal mapping of $r$ at $x \in \mathbb{R}^d$ is

$$\text{prox}_r(x) := \arg\min_{z \in \mathbb{R}^d} r(z) + \frac{\|z - x\|^2}{2}.$$

### 3 Algorithmic Framework

In this section, we propose a unified framework for accelerating the adaptive methods. This framework is composed of two subroutines: Simple Adaptive Subroutine (SAS) and Accelerated Adaptive Subroutine (AAS). Specifically, the framework starts with SAS, which terminates as soon as one successful iteration is identified. Then, the output of SAS is used as the initial point to run AAS until a sufficient number of successful iterations $T_2$ are observed. The details of our algorithmic framework are summarized in Algorithm 1 (in the order of “Main Procedure”, “SAS” and “AAS”).
Algorithm 1: A Generic Unified Adaptive Acceleration Framework (UAA)

Main Procedure:

Input: $x_0 \in \mathbb{R}^d$, $\sigma_0 \geq \sigma_{\min} > 0$, $\tau_0 > 0$, $\gamma_2 > \gamma_1 > 1$, $\gamma_3 > 1$, $\eta > 0$ and $T_2 > 0$.

Phase I (SAS): $[x_0^{AAS}, \sigma_0^{AAS}] = \text{SAS}(x_0, \sigma_0, \sigma_{\min}, \gamma_1, \gamma_2)$.

Phase II (AAS): $[x_{\text{out}}] = \text{AAS}(x_0^{AAS}, \sigma_0^{AAS}, \sigma_{\min}, \tau_0, \gamma_1, \gamma_2, \gamma_3, \eta, T_2)$.

Output: an $\varepsilon$-optimal solution $x_{\text{out}}$.

Simple Adaptive Subroutine: SAS $(x_0, \sigma_0, \sigma_{\min}, \gamma_1, \gamma_2)$

Initialization: the total iteration count $i = 0$ and successful iteration count $j = 0$.

repeat

\begin{itemize}
  \item construct the approximate model $m(x; x_i, \sigma_i)$ and compute $x_{i+1} \approx \argmin_{x \in \mathbb{R}^d} m(x; x_i, \sigma_i)$.
  \item if $F(x_{i+1}) - m(x_{i+1}; x_i, \sigma_i) < 0$ then
    \begin{itemize}
      \item update $\sigma_{i+1} \in [\sigma_{\min}, \sigma_i]$ and $j = j + 1$.
    \end{itemize}
  \item else
    \begin{itemize}
      \item update $x_{i+1} = x_i$ and $\sigma_{i+1} \in [\gamma_1 \sigma_i, \gamma_2 \sigma_i]$.
    \end{itemize}
\end{itemize}

end if

update $i = i + 1$.

until the successful iteration count $j = 1$.

Output: the total number of iterations $i$, the iterate $x_i$ and the regularization parameter $\sigma_i$.

Accelerated Adaptive Subroutine: AAS $(x_0, \sigma_0, \sigma_{\min}, \tau_0, \gamma_1, \gamma_2, \gamma_3, \eta, T_2)$

Initialization: the total iteration count $i = 0$ and successful iteration count $j = 0$.

Initial Step: construct the auxiliary model $\psi_0(z, \tau_0) = l_0(z) + \tau_0 R(z)$, update $\bar{x}_0 = x_0$, compute $z_0 = \arg\min_{z \in \mathbb{R}^d} \psi_0(z, \tau_0)$ and $y_0 = \frac{1}{p+2} \bar{x}_0 + \frac{p+1}{p+2} z_0$.

for $i = 0, 1, 2, \ldots, T_2$ do

\begin{itemize}
  \item construct the approximate model $m(x; y_i, \sigma_i)$.
  \item compute $x_{i+1} \approx \arg\min_{x \in \mathbb{R}^d} m(x; y_i, \sigma_i)$ and $\xi_{i+1} \in \partial r(x_{i+1})$.
  \item if $\theta(x_{i+1}, y_i, \xi_{i+1}) \geq \eta$ then
    \begin{itemize}
      \item update $\bar{x}_{j+1} = x_{i+1}$ and $\xi_{j+1} = \xi_{i+1}$.
      \item update $l_{j+1}(z) = l_j(z) + \Delta l_j(z; \bar{x}_{j+1}, \xi_{j+1})$ and $\tau_{j+1} = \tau_j$.
    \end{itemize}
    \item repeat
      \begin{itemize}
        \item update $\tau_{j+1} = \gamma_3 \tau_{j+1}$, and $z_{j+1} = \arg\min_{z \in \mathbb{R}^d} \{ \psi_{j+1}(z, \tau_{j+1}) = l_{j+1}(z) + \tau_{j+1} R(z) \}$.
        \item update $y_{j+1} = \frac{(j+1)+1}{(j+1)+p+2} \bar{x}_{j+1} + \frac{p+1}{(j+1)+p+2} z_{j+1}$, $\sigma_{i+1} \in [\sigma_{\min}, \sigma_i]$ and $j = j + 1$.
      \end{itemize}
  \end{itemize}
end if

end for

Output: the total number of iterations $i$ and the iterate $x_i$.

We remark that the two-phase scheme is necessary in our analysis to establish the accelerated rate of convergence. Even for the cubic regularization method, without Phase I and exact Hessian information, the proposed method in [19] fails to retain the superior theoretical convergence rate. Some key ingredients of the framework are explained below:

8
Input: The input contains eight parameters: \( x_0 \in \mathbb{R}^d \) is the initial point; \( \sigma_0 \) is the initial regularization parameter for the approximate model; \( \sigma_{\text{min}} \) is the safeguard level for the regularization parameter; \( \tau_0 \) is the initial regularization parameter for the auxiliary model; \( \gamma_1, \gamma_2, \gamma_3 \in (1, +\infty) \) are the ratios for adapting \( \sigma \) and \( \tau \), \( \eta > 0 \) is the threshold for AAS, and \( T_2 \) is the total number of iterations to be run in AAS. Once given the tolerance \( \varepsilon \) of the optimality, we can determine
\[
T_2 = \left( 1 + \frac{2}{\log(\gamma_1)} \log\left( \frac{\sigma_2}{\sigma_{\text{min}}} \right) \right) \left[ 1 + \left( \frac{C}{\varepsilon} \right)^{r^2} \right] \text{ according to Lemma A.2}
\]

Approximativeness measure: Based on the effective approximation \( \overline{m}(y; x) \), we solve the following \((p + 1)\)-th powered regularized function in each iteration of our algorithm:
\[
\min_{y \in \mathbb{R}^d} \ m(y; x, \sigma) = \overline{m}(y; x) + \frac{\sigma}{p + 1} \left\| y - x \right\|^{p+1} + r(y). \tag{8}
\]
In fact, we do not need to solve the above problem exactly but seek an approximate solution, which is defined as follows.

Definition 3.1 Let us call \( \bar{x} \approx \arg\min_{y \in \mathbb{R}^d} m(y; x, \sigma) \) with \( \bar{\xi} \in \partial r(\bar{x}) \) if \( m(\bar{x}; x, \sigma) \leq m(x; x, \sigma) \) and
\[
\left\| \nabla \overline{m}(\bar{x}; x) + \sigma \left\| \bar{x} - x \right\|^{p-1} (\bar{x} - x) + \bar{\xi} \right\| \leq \kappa_\theta \left\| \bar{x} - x \right\|^p, \quad \kappa_\theta > 0. \tag{9}
\]
Note that similar condition of (9) for smooth nonconvex optimization has already been proposed in [11], and in the case \( r(x) = 0 \) and \( p = 2 \), our approximativeness measure does not include the following condition:
\[
(\bar{x} - x)^{\top} \nabla f(x) + (\bar{x} - x)^{\top} \nabla^2 f(x)(\bar{x} - x) + \sigma \left\| \bar{x} - x \right\| \geq 0, \tag{10}
\]
and thus weaker than the one used in [11]. This relaxation also suggests other approximations and implementable solution methods for [10]. For instance, Carmon and Duchi proposed to use gradient descent method to solve [10], and they proved that it works well even when \( m(y; x, \sigma) \) is nonconvex. However, the function \( m(y; x, \sigma) \) in our case is strongly convex, and thus the gradient descent is expected to exhibit a fast (linear) convergence behavior. For more general case of \( r(x) \neq 0 \) and \( p \geq 2 \), we may resort to the accelerated proximal gradient method (APGD) with the initialization \( x_0 = x \) and the step size \( \alpha > 0 \) with the \( k \)-th iteration being:
\[
x_{i,k+1} = \text{prox}_{r/\alpha} \left( x_{i,k} - \frac{\nabla \overline{m}(x_{i,k}; x) + \sigma \left\| x_{i,k} - x \right\|^{p-1} (x_{i,k} - x)}{\alpha} \right),
\]
until \( x_{i,k} \approx \arg\min_{y \in \mathbb{R}^d} m(y; x, \sigma) \).

Auxiliary model: In the framework, we update \( z_{j+1} \) by solving an auxiliary problem:
\[
\arg\min_{z \in \mathbb{R}^d} \left\{ \psi_{j+1}(z, \tau_{j+1}) = l_{j+1}(z) + \tau_{j+1} R(z), \right\}
\]
where \( R(z) = \frac{1}{2(p+1)} \left\| z - x_0 \right\|^{p+1}, l_0(z) = F(x_0), l_{j+1}(z) = l_j(z) + \Delta l_j(z; x_{j+1}, \xi_{j+1}), \) and
\[
\Delta l_j(z, x, \xi) = \frac{\Pi_{\ell=2}^{p+1} (j + \ell)}{p!} \left[ F(x) + (z - x)^{\top} \nabla f(x) + \xi \right].
\]
The auxiliary function $\psi_{j+1}(z, \tau_{j+1})$ is similar to that used by Nesterov in [36, 39] except we need to tune $\tau_{j+1}$ dynamically in the algorithm. This function is the bridge for the two-sided inequality in (25) to establish the iteration bound. In fact, the above subproblem can be solved exactly. To see this, write out the optimality condition and get:

$$\nabla l_{j+1}(z_{j+1}) + \tau_{j+1} \|z_{j+1} - \bar{x}_0\|^{p-1}(z_{j+1} - \bar{x}_0) = 0,$$

which implies that

$$\|z_{j+1} - \bar{x}_0\| = \left(\frac{2\|\nabla \ell_{j+1}(z_{j+1})\|}{\tau_{j+1}}\right)^{1/p}.$$

Moreover, we observe that $l_{j+1}(z)$ is a certain linear function of $z$ and hence $\nabla \ell_{j+1}(z_{j+1})$ is independent of $z_{j+1}$. Consequently, we conclude that

$$z_{j+1} = \bar{x}_0 - \left(\frac{2}{\tau_{j+1}}\right)^{1/p} \frac{\nabla l_{j+1}(z_{j+1})}{\|\nabla l_{j+1}(z_{j+1})\|^{1-1/p}}.$$

**Criterion:** The criterion for determining the successful iteration in AAS is $\theta(x_{i+1}, y_j, \xi_{i+1}) \geq \eta$. In particular, for $p \geq 1$ we define $\theta(x, y, \xi)$ as

$$\theta(x, y, \xi) = \frac{(y - x)\top(\nabla f(x) + \xi)}{\|y - x\|^{p+1}}.$$

**Output:** The output contains the total number of iterations $i$ and the iterate $x_i$. Note that $x_i$ is an $\varepsilon$-optimal solution for problem (1).

### 3.1 Iteration Complexity of the UAA

In this subsection, we first make the following assumption.

**Assumption 3.1** Suppose $x_0$ is the starting point of our algorithm and $x^*$ is an optimal solution of problem (1). The level set $\mathcal{L}(x_0, \sigma) := \{x \in \mathbb{R}^d \mid m(x; x_0, \sigma) \leq m(x_0; x_0, \sigma) = F(x_0)\}$ of $m(\cdot)$ at $x_0$ with regularization parameter $\sigma$ is bounded when $\sigma = \sigma_{\min}$, and that

$$\max_{x \in \mathcal{L}(x_0, \sigma_{\min})} \|x - x^*\| \leq D < \infty.$$  \hspace{1cm} (11)

Then we present the main theoretical results on the iteration complexity of UAA.

**Theorem 3.2** Let the sequence of iterates $\{\bar{x}_j, j \geq 0\}$ be generated by AAS in UAA and $x^*$ be an optimal solution for (1). Denote

$$C := (p + 1)!\left(\frac{2(p + 1)\kappa_p + 2\bar{\sigma}_1 + \tau_0}{2(p + 1)}D^{p+1} + \bar{\kappa}_p D^p + \kappa_{\theta}(2D)^{p+1}\right).$$
Then it holds that
\[ F(\tilde{x}_j) - F(x^*) \leq \frac{C}{\Pi_{i=1}^{p+1}(j + \ell)}, \]
which implies that the total iteration number required to reach \( \varepsilon \)-optimal solution can be bounded by
\[
j \leq 2 + \frac{2}{\log(\gamma_1)} \log \left( \frac{\bar{\sigma}_1}{\sigma_{\min}} \right) + \left[ \frac{1}{\log(\gamma_3)} \log \left( \frac{2(\rho_p + \bar{\sigma}_1 + \kappa_\theta)^p}{\eta^p(p-1)!\tau_0} \right) \right] \\
+ \left( 1 + \frac{2}{\log(\gamma_1)} \log \left( \frac{\sigma_2}{\sigma_{\min}} \right) \right) \left[ 1 + \left( \frac{C}{\varepsilon} \right)^{1+p} \right]. \]

The proof of the theorem is technically involved, thus is postponed to the appendix. To give a holistic picture, we schematically sketch the main steps of the proof below.

**Proof Outline:**

1. We denote \( T_1 \) to be the total number of iterations in SAS. Note that the criterion for the successfully iteration in SAS will be satisfied when \( \sigma_i \) is sufficiently large. Then \( T_1 \) is bounded above by some constant (Lemma A.1).
2. We denote \( T_2 \) by the total number of iterations in AAS, and
   \[ S = \{ j \leq T_2 : j \text{ is a successful iteration in AAS} \} \]
   to be the index set of all successful iterations in AAS. Then \( T_2 \) is bounded above by \( |S| \) multiplied by some constant (Lemma A.2).
3. We denote \( T_3 \) by the total number of counts successfully updating \( \tau > 0 \), and \( T_3 \) is upper bound by some constant (Lemma A.6).
4. We relate the objective function to the count of successful iterations in AAS (Theorem A.8).
5. Putting all the pieces together, we obtain an iteration complexity result (Theorem 3.2).

### 4 Specializations of the UAA

In this section, we provide some concrete choices of \( m(y; x) \), which leads to different iteration complexities of the corresponding algorithms.

#### 4.1 First-Order Adaptive Accelerating Method

The most popular choice of \( m(y; x) \) is the first order approximation:
\[
\bar{m}(y; x) = \hat{f}_1(y; x) = f(x) + (y - x)\top \nabla f(x). 
\]
Obviously, it is convex, and by (3) and (4), (i) and (ii) in Definition 5 are satisfied with
\[ \kappa_1 = \beta_1 = \frac{L_1}{2}, \quad \rho_1 = L_1, \quad \bar{\kappa}_1 = 0. \]

Moreover, the subproblem becomes \( \min_{y \in \mathbb{R}^d} f(x) + (y - x)^T \nabla f(x) + \frac{\sigma}{2} \| y - x \|^2 + r(y) \), which has a closed form solution since \( r(\cdot) \) has an easy proximal mapping. Therefore, \( \kappa_\theta = 0 \) and we have the following iteration bound.

**Theorem 4.1** Letting \( \pi(y; x) = \tilde{f}_1(y; x) \) in UAA, we obtain an adaptive accelerating first-order method, and the total iteration number of getting an \( \epsilon \)-optimal solution is
\[
2 + \frac{2}{\log(\gamma_1)} \log \left( \frac{\bar{\sigma}}{\sigma_{\min}} \right) + \left[ \frac{1}{\log(\gamma_3)} \log \left( \frac{2(L_1^2 + \sigma_2^2)}{\eta \tau_0} \right) \right] + \left( 1 + \frac{2}{\log(\gamma_1)} \log \left( \frac{\sigma_2}{\sigma_{\min}} \right) \right) \left[ 1 + \left( \frac{C_1}{\epsilon} \right)^{\frac{1}{3}} \right]
\]
where
\[ C_1 = \frac{2L_1 + 2\bar{\sigma} + \tau_0}{2} D^2. \]

**4.2 Second-Order Adaptive Accelerating Method**

**4.2.1 Exact Hessian Approximation**

The second order approximation of \( f \) under exact Hessian is given by
\[ \pi(y; x) = \tilde{f}_2(y; x) = f(x) + (y - x)^T \nabla f(x) + \frac{1}{2} (y - x)^T \nabla^2 f(x) (y - x). \]

It is still a convex function. Moreover, by (3) and (4), (i) and (ii) in Definition 5 are satisfied with
\[ \kappa_2 = \beta_2 = \frac{L_2}{6}, \quad \rho_2 = \frac{L_2}{2}, \quad \bar{\kappa}_2 = 0. \]

Moreover, since \( \nabla \pi(y; x) = \nabla^2 f(x) \succeq 0 \), \( \pi(y; x) \) is a convex function. Therefore, we have the following iteration bound.

**Theorem 4.2** Letting \( \pi(y; x) = \tilde{f}_2(y; x) \) in UAA, we obtain an adaptive accelerating cubic regularized Newton’s method, and the total iteration number of getting an \( \epsilon \)-optimal solution is
\[
2 + \frac{2}{\log(\gamma_1)} \log \left( \frac{\bar{\sigma}}{\sigma_{\min}} \right) + \left[ \frac{1}{\log(\gamma_3)} \log \left( \frac{4(L_2^2 + \sigma_2^2 + \kappa_\theta^3)}{\eta^2 \tau_0} \right) \right] + \left( 1 + \frac{2}{\log(\gamma_1)} \log \left( \frac{\sigma_2}{\sigma_{\min}} \right) \right) \left[ 1 + \left( \frac{C_2}{\epsilon} \right)^{\frac{1}{3}} \right]
\]
where
\[ C_2 = 6 \left( \frac{L_2 + 2\bar{\sigma} + \tau_0}{6} D^3 + 8\kappa_\theta D^3 \right). \]
4.2.2 Inexact Hessian Approximation

We study the scenario where the Hessian information is possibly not available; instead, we can construct an approximation of the Hessian $\nabla^2 f(x_i)$ by first computing $d$ forward gradient differences at $x_i$ with a step size $h_i \in \mathbb{R}$,

$$A_i = \left[ \frac{\nabla f(x_i + h_i e_1) - \nabla f(x_i)}{h_i}, \ldots, \frac{\nabla f(x_i + h_i e_d) - \nabla f(x_i)}{h_i} \right],$$

and symmetrizing the resulting matrix: $\tilde{H}(x_i) = \frac{1}{2} (A_i + A_i^T)$ and then further adding a sufficiently large constant multiple of identity matrix to $\tilde{H}(x_i)$: $H(x_i) = \tilde{H}(x_i) + \kappa_c h_i I$, where $e_j$ is the $j$-th vector of the canonical basis. It is well known in [41] that, for some constant $\kappa_c > 0$, we have

$$\left\| \tilde{H}(x_i) - \nabla^2 f(x_i) \right\| \leq \kappa_c h_i.$$

Consequently, it holds that

$$\left\| H(x_i) - \nabla^2 f(x_i) \right\| \leq (\kappa_e + \kappa_c) h_i.$$  

That is to say, the gap between exact and inexact Hessian can be bounded by a multiple of the step size $h_i$. This together with Algorithm 4.1 in [14] motivates a procedure (see the first version of the paper [23]) for searching a pair of $(h_i, x_{i+1})$ such that, $h_i \leq \kappa_{hs} \|x_{i+1} - x_i\|$ for some $\kappa_{hs} > 0$. Therefore, letting $\kappa = (\kappa_e + \kappa_c) \kappa_{hs}$, we conclude that

$$\left\| H(x_i) - \nabla^2 f(x_i) \right\| \leq \kappa \|x_{i+1} - x_i\|.$$  \hspace{1cm} (12)

Therefore, we set

$$m(y; x) = f(x) + (y - x)^\top \nabla f(x) + \frac{1}{2} (y - x)^\top H(x)(y - x)$$

as the second order approximation of $g$ under the inexact Hessian. It follows from [3], [4] and (12) that

$$|f(x_{i+1}) - m(x_{i+1}; x_i)|$$

$$\leq |f(x_{i+1}) - f_2(x_{i+1}; x_i)| + \frac{1}{2} (x_{i+1} - x_i)^\top \nabla^2 f(x_i)(x_{i+1} - x_i) - \frac{1}{2} (x_{i+1} - x_i)^\top H(x_i)(x_{i+1} - x_i)$$

$$\leq \frac{L_2}{6} \|x_{i+1} - x_i\|^3 + \frac{\kappa}{2} \|x_{i+1} - x_i\|^3 = \frac{L_2 + 3\kappa}{6} \|x_{i+1} - x_i\|^3,$$

and

$$|\nabla f(x_{i+1}) - \nabla m(x_{i+1}; x_i)|$$

$$\leq |\nabla f(x_{i+1}) - \nabla f_2(x_{i+1}; x_i)| + |\nabla^2 f(x_i)(x_{i+1} - x_i) - H(x_i)(x_{i+1} - x_i)|$$

$$\leq \frac{L_2}{2} \|x_{i+1} - x_i\|^2 + \kappa \|x_{i+1} - x_i\|^2 = \frac{L_2 + 2\kappa}{2} \|x_{i+1} - x_i\|^2.$$

Moreover, since $\nabla f(x)$ is Lipschitz continuous with $L_1 > 0$, we have

$$\|\nabla^2 f(x)\| \leq L_1, \quad x \in \text{dom}(F).$$
As a result
\[
|f(y) - \overline{m}(y; x)| \\
\leq |f(y) - f_2(y; x)| + \frac{1}{2}(y - x)^\top \nabla^2 f_2(x)(y - x) - \frac{1}{2}(y - x)^\top H(x)(y - x) \\
\leq \frac{L_2}{6} \|y - x\|^3 + \frac{1}{2} \|y - x\|^2 (\|\nabla^2 f_2(x)\| + \|H(x)\|) \\
\leq \frac{L_2}{6} \|y - x\|^3 + L_1 \|y - x\|^2.
\]

Finally, since \( f \) is convex and \( \kappa_c \) is sufficiently large such that \( \kappa_c \geq \kappa_e \), we have
\[
H(x_i) \succeq \nabla^2 f(x_i) - \kappa_e h_i I + \kappa_c h_i I \succeq 0,
\]
and \( \overline{m}(y; x_i) \) is convex as well. Therefore, all three conditions in Definition 5 are satisfied with
\[
\beta_2 = \frac{L_2 + 3\kappa}{6}, \quad \rho_2 = \frac{L_2 + 2\kappa}{2}, \quad \kappa_2 = \frac{L_2}{6}, \quad \bar{\kappa}_2 = L_1.
\]
Therefore, we have the following iteration bound.

**Theorem 4.3** Letting \( \overline{m}(y; x) = f(x) + (y - x)^\top \nabla f(x) + \frac{1}{2}(y - x)^\top H(x)(y - x) \) in UAA, we obtain an adaptive accelerating cubic regularized approximate Newton’s method, and the total iteration number of getting an \( \epsilon \)-optimal solution is
\[
2 + \frac{2\log\left(\frac{\sigma_1}{\sigma_{\text{min}}}\right)}{\log(\gamma_1)} + \left[ \frac{1}{\log(\gamma_3)} \log\left( \frac{4(L_2 + 2\kappa + \bar{\sigma}_2 + \kappa_\theta)^3}{\eta^2 \tau_0} \right) \right] + \left[ 1 + \frac{2\log(\gamma_1)}{\log(\gamma_3)} \log\left( \frac{\bar{\sigma}_2}{\sigma_{\text{min}}} \right) \right] \left[ 1 + \left( \frac{\bar{C}_2}{\epsilon} \right)^{\frac{1}{3}} \right],
\]
where
\[
\bar{C}_2 = 6 \left( \frac{L_2 + 2\bar{\sigma}_1 + \tau_0}{6} D^3 + L_1 D^2 + 8\kappa_\theta D^3 \right).
\]

### 4.3 High-Order Adaptive Accelerating Method

To utilize high-order information, we let
\[
\overline{m}(y; x) = \tilde{f}_p(y; x).
\]
Then by invoking (3) and (4), (i) and (ii) in Definition 5 are satisfied with
\[
\kappa_p = \beta_p = \frac{L_p}{(p + 1)!}, \quad \beta_p = \frac{L_p}{p!}, \quad \bar{\kappa}_p = 0.
\]
Unfortunately, \( \overline{m}(y; x) \) is not necessarily convex in this case. However, according to Theorem 1 in [39],
\[
\overline{m}(y; x) + \sigma \frac{||y - x||^{p+1}}{p + 1}
\]
is a convex function when $\sigma \geq L_p(p-1)!$. Therefore,

$$m(y; x, \sigma) = m(y; x) + \frac{\sigma \|y - x\|^{p+1}}{p+1} + r(y)$$

is convex when $\sigma \geq L_p(p-1)!$. Recall that the convexity of $m(y; x, \sigma)$ at point $\bar{x}_0$ is only used in the proof of Theorem A.7 to get an upper bound of $\psi_0(z, \tau_0)$. Generally speaking, checking the convexity of a polynomial function could be very hard [2], and it remains a challenging task even when such polynomial equipped with some structure (for instance, sum of squares) [22]. However, checking the convexity at one point is computationally tractable, which boils down to verifying the positive semidefiniteness of the Hessian matrix. This motivates us to add checking if $m(x_{i+1}; x_i, \sigma_i) = m(x_{i+1}; x_i, \sigma_i)$ is convex as another successful criterion in SAS, in addition to the original one $F(x_{i+1}) \geq m(x_{i+1}; x_i, \sigma_i)$. Therefore, when

$$\sigma_i \geq \max\{(p+1)\beta_p, L_p(p-1)\!\right) = \max\left\{\frac{L_p}{p!}, L_p(p-1)!\right\} = L_p(p-1)!,$$

both criteria are satisfied. Therefore, we only need to modify

$$\bar{\sigma}_1 = \max\{\sigma_0, \gamma_2 L_p(p-1)\!\right),$$

and keep the other parameters unchanged. Therefore, we have the following iteration bound.

**Theorem 4.4** Letting $\tilde{\pi}(y; x) = \tilde{f}_p(y; x)$ in UAA, we obtain an adaptive accelerating $p$-th order method, and the total iteration number of getting an $\epsilon$-optimal solution is

$$2 + \frac{2}{\log(\gamma_1)} \log \left( \frac{\bar{\sigma}_1}{\sigma_{\min}} \right) + \left[ \frac{1}{\log(\gamma_3)} \log \left( \frac{2}{(p+1)!} + \bar{\sigma}_2 + \kappa_\theta \right)^{p+1} \right] \eta^p(p-1)! \tau_0^{p-1}$$

$$+ \left( 1 + \frac{2}{\log(\gamma_1)} \log \left( \frac{\bar{\sigma}_2}{\sigma_{\min}} \right) \right) \left[ 1 + \left( \frac{C_p}{\epsilon} \right)^{p+1} \right]$$

where

$$C_p = (p+1)! \left( \frac{2L_p}{p!} + 2\bar{\sigma}_1 + \tau_0 \right) \frac{D^{p+1}}{2(p+1)} + \kappa_\theta (2D)^{p+1}.$$

### 5 Numerical Experiments

In this section, we present the results of some numerical experiments for solving the $\ell_1/\ell_2$-regularized logistic regression problems.

#### 5.1 $\ell_2$-Regularized Logistic Regression Problem

We first test the performance of the algorithms by evaluating the following $\ell_2$-regularized logistic regression problem

$$\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^{n} \ln \left( 1 + \exp \left( -b_i \cdot a_i^T x \right) \right) + \frac{\lambda}{2} \|x\|^2$$

(13)
where \((a_i, b_i)_{i=1}^n\) is the samples in the data set, and the regularization parameter is set as \(\lambda = 10^{-5}\). To observe the acceleration, the starting point is randomly generated from a Gaussian random variable with zero mean and a large variance (say 5000). In this way, initial solutions are likely to be far away from the global solution.

We implement a variant of Algorithm 1 with cubic regularization, referred to as *Adaptively Accelerated Cubic Regularized* (AARC) Newton’s method. In this variant we first run Algorithm 1. After 10 successful iterations of *Accelerated Adaptive Subroutine* are performed, we check the progress made by each iteration. In particular, when \(\frac{|f(x_{k+1}) - f(x_k)|}{|f(x_k)|} \leq 0.1\), which indicates that it is getting close to the global optimum, then we switch to the adaptive cubic regularization phase of Newton’s method (ARC) in [11, 12] with stopping criterion \(\|\nabla f(x)\| \leq 10^{-9}\). In the implementation, we apply the so-called Lanczos process to approximately solve the subproblem \(\min_{y \in \mathbb{R}^d} m(y; x_i, \sigma_i)\). In addition to (9), the approximate solution \(s\) is also made to satisfy

\[
(y - x_i)^T \nabla f(x_i) + (y - x_i)^T \nabla^2 f(x_i)(y - x_i) + \sigma \|y - x_i\|^3 = 0
\]

for given \(x_i\) and \(\sigma_i\). Note that (14) is a consequence of the first order necessary condition, and as shown in Lemma 3.2 [11], the global minimizer of \(m(y; x_i, \sigma_i)\) when restricted to a Krylov subspace

\[K := \text{span}\{\nabla f(x_i), \nabla^2 f(x_i)\nabla f(x_i), (\nabla^2 f(x_i))^2 \nabla f(x_i), \ldots\}\]

does not depend on the subspace dimension. Moreover, minimizing \(m(y; x_i, \sigma_i)\) in the Krylov subspace only involve factorizing a tri-diagonal matrix, which can be done at the cost of \(O(d)\). Thus, the associated approximate solution can be found through the so-called Lanczos process, where the dimension of \(K\) is gradually increased and an orthogonal basis of each subspace \(K\) is built up which typically involves one matrix-vector product. Condition (9) can be used as the termination criterion for the Lanczos process in the hope to find a suitable trial step before the dimension of \(K\) approaches \(d\).

| Dataset | \(n\) | \(d\) |
|---------|------|------|
| sonar   | 208  | 60   |
| splice  | 1,000| 60   |
| svmguid1 | 3,089| 4    |
| svmguid3 | 1,243| 22   |
| w8a     | 49,749| 300  |
| SUSY    | 5,000,000| 18  |

Table 2: Statistics of datasets for regularized logistic regression.

We compare the new AARC method with 5 other methods, including the adaptive cubic regularization of Newton’s method (ARC), the trust region method (TR), the limited memory Broyden-Fletcher-Goldfarb-Shanno method (L-BFGS) that is implemented in SCIPY Solvers\(^1\). Algorithm 1 with quadratic regularization referred to as adaptive accelerated gradient descent (AAGD) and the standard Nesterov’s accelerated gradient descent (AGD). The experiments are conducted on 6 LIBSVM Sets\(^2\) for binary classification, and the summary of those datasets are shown in Table 2.

The results in Figure 1 and Figure 2 confirm that AARC indeed accelerates ARC, especially when the current iterate has not entered the local region of quadratic convergence yet. Moreover, AARC

\(^1\)https://docs.scipy.org/doc/scipy/reference/optimize.html#module-scipy.optimize

\(^2\)https://www.csie.ntu.edu.tw/~cjlin/libsvm/
5.2 $\ell_1$-Regularized Logistic Regression Problem

Then we test the algorithms on the following $\ell_1$-regularized logistic regression problem:

$$
\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \log \left( 1 + \exp(-y_i w_i^T x) \right) + \lambda \|x\|_1, 
$$

(15)

where $\{(w_i, y_i)\}_{i=1}^{n}$ is a collection of data samples with $y_i \in \{-1, 1\}$ being the label. The regularization term $\|x\|_1$ promotes sparse solutions and $\lambda > 0$ balances sparsity with goodness-of-fit and generalization. In addition, $\lambda$ was chosen by LIBLINEAR with five-fold cross validation. The experiments are conducted on 3 data sets that all come from LIBSVM\footnote{The LIBSVM collection is available at \url{https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets}}, and the summary of those datasets are shown in Table 3.
Figure 2: Performance of AARC and all benchmark methods on the task of $\ell_2$-regularized logistic regression (loss vs. iterations)

| Name   | Description       | $n$  | $d$  | Scaled Interval | $\lambda$ |
|--------|-------------------|------|------|-----------------|-----------|
| a9a    | UCI adult         | 48842| 123  | [0, 1]          | 4.5e-03   |
| covtype| forest covtype    | 581012| 54   | [0, 1]          | 2.6e-03   |
| w8a    |                   | 64700| 300  | [0, 1]          | 7.0e-04   |

Table 3: Statistics of datasets for $\ell_1$-regularized logistic regression.

We compare an accelerated proximal cubic regularization of Newton method with the TFOCS\(^3\) implementation of Nesterov’s accelerated gradient method (Nesterov83) and fast iterative shrinkage-thresholding algorithm (FISTA) on $\ell_1$-regularized logistic regression problem (15). The subproblem solver is the gradient descent method which has been proven convergent for cubic regularized polynomial [10]. We plot relative suboptimality versus iteration counts as well as relative suboptimality versus time on all datasets in Figure 3. It is clear in Figure 3 that our method consistently performs the best in terms of the number of iterations and the overall computational time.

\(^3\)http://cvxr.com/tfocs/
Figure 3: Iteration counts and consuming of three methods on $\ell_1$-regularized logistic regression.
| Algorithms | log($\|\nabla f\|$) | 0.5 | 0.1 | 1e-2 | 1e-4 | 1e-6 | 1e-8 |
|------------|----------------|-----|-----|------|------|------|------|
| sonar      |                |     |     |      |      |      |      |
| AARC       | 19             | 194 | 551 | 564  | 565  | 567  |
| ARC        | 28             | 175 | 673 | 705  | 707  | 708  |
| TR         | 24             | 102 | 438 | 524  | 525  | 527  |
| L-BFGS     | 7              | 19  | 183 | *    | *    | *    |
| splice     |                |     |     |      |      |      |      |
| AARC       | 131            | 655 | 785 | 786  | 787  | 787  |
| ARC        | 118            | 669 | 841 | 842  | 843  | 843  |
| TR         | 128            | 523 | 621 | 622  | 623  | 624  |
| L-BFGS     | 16             | 77  | 95  | 135  | 165  | *    |
| svmguide1  |                |     |     |      |      |      |      |
| AARC       | 166            | 511 | 628 | 629  | 629  | 630  |
| ARC        | 200            | 787 | 1118| 1118 | 1119 | 1119 |
| TR         | 175            | *   | *   | *    | *    | *    |
| L-BFGS     | 16             | 44  | 45  | 46   | *    | *    |
| svmguide3  |                |     |     |      |      |      |      |
| AARC       | 1              | 34  | 91  | 98   | 100  | 104  |
| ARC        | 1              | 77  | 101 | 108  | 110  | 112  |
| TR         | 1              | 54  | 120 | 133  | 134  | 136  |
| L-BFGS     | 1              | 11  | 52  | 252  | 460  | *    |
| w8a        |                |     |     |      |      |      |      |
| AARC       | 23             | 26  | 79  | 105  | 109  | 110  |
| ARC        | 23             | 26  | 85  | 111  | 114  | 116  |
| TR         | 107            | 117 | 142 | 160  | 162  | 163  |
| L-BFGS     | 3              | 6   | 20  | *    | *    | *    |
| SUSY       |                |     |     |      |      |      |      |
| AARC       | 18             | 37  | 95  | 97   | 99   | 102  |
| ARC        | 18             | 35  | 140 | 143  | 144  | 147  |
| TR         | 17             | 89  | 178 | 181  | 183  | 185  |
| L-BFGS     | 5              | 14  | 86  | 163  | 231  | *    |

Table 4: Iteration counts of various second-order methods to reach different levels of losses.

Acknowledgements. We would like to thank the anonymous referees for their insightful comments, which helped to significantly improve this paper from its original version. We would like to express our
deep gratitude toward Professor Xi Chen of Stern School of Business at New York University for the fruitful discussions at various stages of this project.

References

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

[2] A. A. Ahmadi, A. Olshevsky, P. A. Parrilo, and J. N. Tsitsiklis. NP-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical Programming*, 137(1-2):453–476, 2013.

[3] Z. Allen-Zhu and L. Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *ArXiv Preprint: 1407.1537*, 2014.

[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[5] S. Becker, J. Fadili, and P. Ochs. On quasi-Newton forward–backward splitting: Proximal calculus and convergence. *ArXiv Preprint: 1801.08691*, 2018.

[6] E.G. Birgin, J.L. Gardenghi, J.M. Martinez, S.A. Santos, and Ph.L. Toint. On high-order model regularization for constrained optimization. *Mathematical Programming*, 163(1-2):359–368, 2017.

[7] S. Bubeck, Y. T. Lee, and M. Singh. A geometric alternative to Nesterov’s accelerated gradient descent. *ArXiv Preprint: 1506.08187*, 2015.

[8] R. H. Byrd, J. Nocedal, and F. Oztoprak. An inexact successive quadratic approximation method for $l_1$ regularized optimization. *Mathematical Programming*, 157(2):375–396, 2016.

[9] L. Calatroni and A. Chambolle. Backtracking strategies for accelerated descent methods with smooth composite objectives. [arXiv:1709.09004], 2017.

[10] Y. Carmon and J. Duchi. Gradient descent efficiently finds the cubic-regularized non-convex Newton step. *ArXiv Preprint: 1612.00547v2*, 2016.

[11] 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, 2011.

[12] 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, 2011.

[13] C. Cartis, N. I. M. Gould, and P. L. Toint. Evaluation complexity of adaptive cubic regularization methods for convex unconstrained optimization. *Optimization Methods and Software*, 27(2):197–219, 2012.
[14] C. Cartis, N. I. M. Gould, and P. L. Toint. On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. *SIAM Journal on Optimization*, 22(1):66–86, 2012.

[15] C. Cartis, N.I.M. Gould, and Ph.L. Toint. Improved second-order evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *arXiv:1708.04044*, 2017.

[16] A. Cotter, O. Shamir, N. Srebro, and K. Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *NIPS*, pages 1647–1655, 2011.

[17] Y. Drori and M. Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482, 2014.

[18] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7):2121–2159, 2011.

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

[20] H. Ghanbari and K. Scheinberg. Proximal quasi-newton methods for regularized convex optimization with linear and accelerated sublinear convergence rates. *Computational Optimization and Applications*, 69(3):597–627, 2018.

[21] G.N. Grapiglia and Y. Nesterov. Accelerated regularized newton methods for minimizing composite convex functions. *Technical Report CORE Discussion paper*, 2018.

[22] B. Jiang, Z. Li, and S. Zhang. On cones of nonnegative quartic forms. *Foundations of Computational Mathematics*, 17(1):161–197, 2017.

[23] B. Jiang, T. Lin, and S. Zhang. A unified scheme to accelerate adaptive cubic regularization and gradient methods for convex optimization. *ArXiv Preprint: 1710.04788*, 2017.

[24] A. Karparthy. A peak at trends in machine learning. *https://medium.com/@karpathy/a-peek-at-trends-in-machine-learning-ab8a1085a106*, 2017.

[25] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *ArXiv Preprint: 1412.6980*, 2014.

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

[27] J. D. Lee, Y. Sun, and M. A. Saunders. Proximal newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1443, 2014.

[28] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *NIPS*, pages 3384–3392, 2015.

[29] Q. Lin and L. Xiao. An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization. *Computational Optimization and Applications*, 60(3):633–674, 2014.
[30] D. G. Luenberger and Y. Ye. *Linear and nonlinear programming*, volume 2. Springer, 1984.

[31] J.M. Martínez. On high-order model regularization for constrained optimization. *SIAM Journal on Optimization*, 27(4):2447–2458, 2017.

[32] R. D. C. Monteiro, C. Ortiz, and B. F. Svaiter. An adaptive accelerated first-order method for convex optimization. *Computational Optimization and Applications*, 64(1):31–73, 2016.

[33] 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.

[34] 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.

[35] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $\mathcal{O}(1/k^2)$. *Doklady AN SSSR, translated as Soviet Math. Docl.*, 269:543–547, 1983.

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

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

[38] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[39] Y. Nesterov. Implementable tensor methods in unconstrained convex optimization. *Technical Report CORE Discussion paper*, 2018.

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

[41] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, 2006.

[42] K. Scheinberg, D. Goldfarb, and X. Bai. Fast first-order methods for composite convex optimization with backtracking. *Foundations of Computational Mathematics*, 14:389–417, 2014.

[43] K. Scheinberg and X. Tang. Practical inexact proximal quasi-newton method with global complexity analysis. *Mathematical Programming*, 160(1-2):495–529, 2016.

[44] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *ICML*, pages 64–72, 2014.

[45] O. Shamir and R. Shiff. Oracle complexity of second-order methods for smooth convex optimization. *ArXiv Preprint: 1705.07260*, 2017.

[46] W. Su, S. Boyd, and E. J. Candes. A differential equation for modeling Nesterovs accelerated gradient method: theory and insights. *Journal of Machine Learning Research*, 17(153):1–43, 2016.
[47] T. Tieleman and G. Hinton. Lecture 6.5-RMSProp: divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012.

[48] A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, pages 7351–7358, 2016.

[49] A. C. Wilson, B. Recht, and M. I. Jordan. A Lyapunov analysis of momentum methods in optimization. *ArXiv Preprint: 1611.02635*, 2016.

[50] P. Xu, F. Roosta-Khorasani, and M. W. Mahoney. Newton-type methods for non-convex optimization under inexact Hessian information. *ArXiv Preprint: 1708.07164*, 2017.
A Technical Proofs in Section 3

Firstly, we bound the total number of iteration in SAS, denoted as $T_1$, and the total number of iteration in AAS, denoted as $T_2$.

**Lemma A.1** Let $\bar{\sigma}_1 = \max \{\sigma_0, (p + 1)\gamma_2 \beta_p\}$.

We have

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

**Proof.** We observe that

$$F(x_{i+1}) = f(x_{i+1}) + r(x_{i+1}) \leq m(x_{i+1}; x_i) + \beta_p ||x_{i+1} - x_i||^{p+1} + r(x_{i+1}) = m(x_{i+1}; x_i, \sigma_i) + \left( \frac{\beta_p}{p+1} \right) ||x_{i+1} - x_i||^{p+1}.$$

Hence

$$\sigma_i \geq (p + 1)\beta_p \Rightarrow F(x_{i+1}) \leq m(x_{i+1}; x_i, \sigma_i).$$

This implies that

$$\sigma_i < (p + 1)\beta_p, \quad \forall i \leq T_1 - 2.$$

Therefore, we conclude that

$$\sigma_{\min} \leq \sigma_{T_1} \leq \sigma_{T_1-1} \leq \max \{\sigma_0, (p + 1)\gamma_2 \beta_p\} = \bar{\sigma}_1.$$

Combining the facts that $\sigma_{\min} \leq \sigma_0 \leq \bar{\sigma}_1$ and $\gamma_1 \sigma_i \leq \sigma_{i+1}$ for all unsuccessful iterations yields that

$$\frac{\bar{\sigma}_1}{\sigma_{\min}} \leq \frac{\sigma_{T_1}}{\sigma_0} \leq \frac{\sigma_{T_1}}{\sigma_{T_1-1}} \cdot \prod_{i=0}^{T_1-2} \frac{\sigma_{i+1}}{\sigma_i} \geq \gamma_1^{T_1-1} \left( \frac{\sigma_{\min}}{\bar{\sigma}_1} \right)$$

which implies the desired result. This completes the proof. \qed

We remark that the above result is motivated from Theorem 2.1 in [12], and our proof is mostly identical to the one in [12].

**Lemma A.2** Let $S$ be the set of successful iteration counts in the total iteration count and

$$\bar{\sigma}_2 = \max \{\bar{\sigma}_1, \gamma_2 (\kappa_{\Theta} + \rho_p + \eta)\}.$$

Then we have

$$T_2 \leq \left( 1 + \frac{2}{\log (\gamma_1)} \log \left( \frac{\bar{\sigma}_2}{\sigma_{\min}} \right) \right) |S|.$$
Proof. We observe that
\[
(y_j - x_{i+1})^\top \left( \nabla m(x_{i+1}; y_j) + \sigma_i \|x_{i+1} - y_j\|^{p-1} (x_{i+1} - y_j) + \xi_{i+1} \right)
\geq - \|y_j - x_{i+1}\| \cdot \left\| \nabla m(x_{i+1}; y_j) + \sigma_i \|x_{i+1} - y_j\|^{p-1} (x_{i+1} - y_j) + \xi_{i+1} \right\| \tag{16}
\]
Consequently, we conclude that
\[
\theta(x_{i+1}, y_j, \xi_{i+1}) = \frac{(y_j - x_{i+1})^\top (\nabla f(x_{i+1}) + \xi_{i+1})}{\|y_j - x_{i+1}\|^{p+1}}
\geq \sigma_i - \kappa \theta - (y_j - x_{i+1})^\top (\nabla f(x_{i+1}) - \nabla m(x_{i+1}; y_j)) \geq \sigma_i - \kappa \theta - (y_j - x_{i+1}) \geq \sigma_i - \kappa \theta - \rho_p \tag{17}
\]
and
\[
\sigma_i \geq \kappa \theta + \rho_p + \eta \implies \theta(x_{i+1}, y_j, \xi_{i+1}) \geq \eta.
\]
This implies that
\[
\sigma_{i+1} \leq \sigma_i \leq \gamma_2 \sigma_{i-1} \leq \gamma_2 (\kappa \theta + \rho_p + \eta), \quad \forall i \in S.
\]
Therefore, \(\sigma_i\) can be upper bounded by \(\hat{\sigma}_2\) and lower bounded by \(\sigma_{\min}\) in AAS. In addition, \(\gamma_1 \sigma_i \leq \sigma_{i+1}\) for any \(i \notin S\). Therefore, we have
\[
\frac{\hat{\sigma}_2}{\sigma_{\min}} = \left( \prod_{i \in S} \frac{\sigma_{i+1}}{\sigma_i} \cdot \prod_{i \notin S} \frac{\sigma_{i+1}}{\sigma_i} \right) \geq \gamma_1 \frac{T_2 - |S|}{\frac{T_2}{\sigma_2}} \left( \frac{\sigma_{\min}}{\sigma_0} \right)^{|S|},
\]
which further implies an upper bound for \(T_2\), completing the proof. \(\square\)

Next we proceed to bounding the total number of times updating the regularization parameter \(\tau\) in the auxiliary model, which is denoted as \(T_3\). This requires three key technical lemmas presented below.

Lemma A.3 For any \(g \in \mathbb{R}^d\), \(s \in \mathbb{R}^d\) and integer \(q \geq 2\), we have
\[
g^\top s + \frac{\sigma \|s\|^q}{q} \geq \frac{q - 1}{q} \left( \frac{\|g\|^q}{\sigma} \right)^{\frac{1}{q-1}}. \tag{17}
\]
Proof. We denote \( s^* \) as the minimizer of \( g^\top s + \frac{\sigma}{q} \| s \|^q \) and the first-order optimality condition gives that \( g + \sigma \| s^* \|^{q-2} s^* = 0 \). This implies that
\[
g^\top s^* + \sigma \| s^* \|^q = 0 \quad \text{and} \quad \| g \| = \sigma \| s^* \|^{q-1},
\]
and hence
\[
g^\top s + \frac{\sigma \| s \|^q}{q} \geq g^\top s^* + \frac{\sigma \| s^* \|^q}{q} = -\frac{(q-1)}{q} \sigma \| s^* \|^q = -\frac{q-1}{q} \left( \frac{\| g \|^q}{\sigma} \right)^{\frac{1}{q-1}}.
\]
\( \square \)

**Lemma A.4** For the minimizer of \( \psi_j(z, \tau_j) \) over \( z \in \mathbb{R}^d \), i.e.,
\[
z_j = \arg\min_{z \in \mathbb{R}^d} \psi_j(z, \tau_j),
\]
and any integer \( q \geq 2 \), we have
\[
\psi_j(z, \tau_j) - \psi_j(z_j, \tau_j) \geq \frac{\tau_j}{2q-1} \frac{\| z - z_j \|^q}{q}.
\]

Proof. Recall that \( \psi_j(z, \tau_j) \) is the sum of a linear function and a \( q \)-th powered regularization function:
\[
\psi_j(z, \tau_j) = l_j(z) + \tau_j R(z) = l_j(z) + \frac{\tau_j}{2} \frac{\| z - x_0 \|^q}{q}.
\]

Thus, we have
\[
\psi_j(z, \tau_j) - \psi_j(z_j, \tau_j) = l_j(z) + \tau_j R(z) - l_j(z_j) - \tau_j R(z_j)
\]
\[
= (z - z_j)^\top \nabla l_j(z_j) + \tau_j (R(z) - R(z_j))
\]
\[
\geq (z - z_j)^\top \nabla l_j(z_j) + \tau_j (z - z_j)^\top \nabla R(z_j) + \frac{\tau_j}{2} \frac{\| z - z_j \|^q}{q^{2q-2}}
\]
\[
= (z - z_j)^\top \nabla l_j(z_j) + \tau_j (z - z_j)^\top \nabla R(z_j) + \frac{\tau_j}{2q-1} \frac{\| z - z_j \|^q}{q}.
\]

Since \( z_j \) is the minimizer of \( \psi_j(z, \tau_j) \) over \( z \in \mathbb{R}^d \), we have
\[
\nabla l_j(z_j) + \tau_j \nabla R(z_j) = \nabla \psi_j(z_j, \tau_j) = 0.
\]
Combining the above two formulas yeilds the desired result. \( \square \)

**Lemma A.5** For any \( j \geq 0 \) in AAS, then we have
\[
\| \nabla f(\bar{x}_{j+1}) + \bar{\xi}_{j+1} \| \leq (\rho_p + \bar{\sigma}_2 + \kappa \theta) \| \bar{x}_{j+1} - y_j \|^p.
\]
Proof. We observe that
\[
\begin{aligned}
\|\nabla m(\bar{x}_{j+1}^1; y_j) + \bar{\xi}_{j+1}\| & \leq \|\nabla m(\bar{x}_{j+1}; y_j) + \sigma_i \|\bar{x}_{j+1} - y_j\|^{p-1} (\bar{x}_{j+1} - y_j) + \sigma_i \|\bar{x}_{j+1} - y_j\|^p \\
& \leq \kappa_\theta \|\bar{x}_{j+1} - y_j\|^p + \sigma_i \|\bar{x}_{j+1} - y_j\|^p \\
& \leq (\kappa_\theta + \bar{\sigma}_2) \|\bar{x}_{j+1} - y_j\|^p.
\end{aligned}
\]

Therefore,
\[
\begin{aligned}
\|\nabla f(\bar{x}_{j+1}) + \bar{\xi}_{j+1}\| & \leq \|\nabla f(\bar{x}_{j+1}) - \nabla m(\bar{x}_{j+1}; y_j)\| + \|\nabla m(\bar{x}_{j+1}; y_j) + \bar{\xi}_{j+1}\| \\
& \leq \rho_p \|\bar{x}_{j+1} - y_j\|^p + (\kappa_\theta + \bar{\sigma}_2) \|\bar{x}_{j+1} - y_j\|^p \\
& = (\rho_p + \bar{\sigma}_2 + \kappa_\theta) \|\bar{x}_{j+1} - y_j\|^p.
\end{aligned}
\]

Completing the proof. □

We remark that the above result is motivated from Lemma 5.2 in [12], which originally works for cubic regularized methods with smooth objective function. Next, we bound \(T_3\), the total number of times updating \(\tau\) in the auxiliary model:

**Lemma A.6** For any \(j \geq 0\) in AAS, we have
\[
\psi_j(z_j, \tau_j) \geq \frac{\prod_{\ell=1}^{p+1}(j + \ell)}{(p + 1)!} F(\bar{x}_j)
\]
providing that
\[
\tau_j \geq \frac{2 (\rho_p + \bar{\sigma}_2 + \kappa_\theta)^{p+1} p^{p-1}}{\eta^p (p - 1)!} > 0.
\]
As a consequence,
\[
T_3 \leq 1 + \left\lfloor \frac{1}{\log(\gamma_3)} \log \left( \frac{2 (\rho_p + \bar{\sigma}_2 + \kappa_\theta)^{p+1} p^{p-1}}{\eta^p (p - 1)! \tau_0} \right) \right\rfloor.
\]

Proof. We shall prove by induction.

First of all, the base case of \(j = 0\) holds true due to the fact that
\[
\psi_0(z_0, \tau_0) = \min_{z \in \mathbb{R}^d} F(\bar{x}_0) + \frac{\tau_0 \|z - \bar{x}_0\|^{p+1}}{2(p + 1)} = F(\bar{x}_0).
\]

Then, we assume the result to hold for some \(j = j_0\). It remains to prove the result for the case \(j = j_0 + 1\). By the induction hypothesis, and by Lemma A.4, we have
\[
\psi_{j_0}(z, \tau_{j_0}) \geq \psi_{j_0}(z_{j_0}, \tau_{j_0}) + \frac{\tau_{j_0} \|z - z_{j_0}\|^{p+1}}{2(p + 1)} \\
\geq \frac{\prod_{\ell=1}^{p+1}(j_0 + \ell)}{(p + 1)!} F(\bar{x}_{j_0}) + \frac{\tau_{j_0} \|z - z_{j_0}\|^{p+1}}{2(p + 1)}.
\]
Furthermore, observe that
\[
\psi_{j_0 + 1}(z_{j_0 + 1}, \tau_{j_0 + 1}) = \min_{z \in \mathbb{R}^d} \psi_{j_0 + 1}(z, \tau_{j_0 + 1})
\]
\[
= \min_{z \in \mathbb{R}^d} \{ l_{j_0 + 1}(z) + \tau_{j_0 + 1}R(z) \}
= \min_{z \in \mathbb{R}^d} \{ l_{j_0}(z) + \Delta l_{j_0}(z; \tilde{x}_{j_0}, \tilde{\xi}_{j_0}) + \tau_{j_0}R(z) + (\tau_{j_0 + 1} - \tau_{j_0})R(z) \}
\geq \min_{z \in \mathbb{R}^d} \left\{ \psi_{j_0}(z, \tau_{j_0}) + \Delta l_{j_0}(z; \tilde{x}_{j_0}, \tilde{\xi}_{j_0}) \right\},
\]
where the last inequality is because of the fact that $\tau_{j_0 + 1} \geq \tau_{j_0}$ and $R(z) \geq 0$ for any $z \in \mathbb{R}^d$, and
\[
\Delta l_{j_0}(z; \tilde{x}_{j_0}, \tilde{\xi}_{j_0}) = \frac{\Pi_{\ell=2}(j_0 + \ell)}{p!} \left[ F(x_{j_0 + 1}) + (z - x_{j_0 + 1})^\top (\nabla f(x_{j_0 + 1}) + \tilde{\xi}_{j_0 + 1}) \right].
\]
Therefore, we have
\[
\psi_{j_0}(z, \tau_{j_0}) + \Delta l_{j_0}(z, \tilde{x}_{j_0 + 1}) = \frac{\Pi_{\ell=2}(j_0 + \ell)}{(p + 1)!} F(\tilde{x}_{j_0 + 1}) + \tau_{j_0} \left\| z - x_{j_0} \right\|^{p+1} + \frac{\Pi_{\ell=2}(j_0 + \ell)}{p!} \left[ F(\tilde{x}_{j_0 + 1}) + (z - \tilde{x}_{j_0 + 1})^\top (\nabla f(\tilde{x}_{j_0 + 1}) + \tilde{\xi}_{j_0 + 1}) \right] + \tau_{j_0} \left\| z - x_{j_0} \right\|^{p+1} + \frac{\Pi_{\ell=2}(j_0 + \ell)}{p!} \left[ F(\tilde{x}_{j_0 + 1}) + (z - \tilde{x}_{j_0 + 1})^\top (\nabla f(\tilde{x}_{j_0 + 1}) + \tilde{\xi}_{j_0 + 1}) \right] + \frac{\Pi_{\ell=2}(j_0 + \ell)}{(p + 1)!} (\tilde{x}_{j_0} - \tilde{x}_{j_0 + 1})^\top (\nabla f(\tilde{x}_{j_0 + 1}) + \tilde{\xi}_{j_0 + 1}) + \frac{\Pi_{\ell=2}(j_0 + \ell)}{p!} (z - \tilde{x}_{j_0 + 1})^\top (\nabla f(\tilde{x}_{j_0 + 1}) + \tilde{\xi}_{j_0 + 1}).
\]
Moreover, $y_{j_0}$ in the algorithm is constructed to satisfy
\[
y_{j_0} = \frac{j_0 + 1}{j_0 + p + 2} \tilde{x}_{j_0} + \frac{p + 1}{j_0 + p + 2} z_{j_0},
\]
and thus
\[
\frac{\Pi_{\ell=1}(j_0 + \ell)}{(p + 1)!} \tilde{x}_{j_0} = \frac{\Pi_{\ell=1}(j_0 + \ell)}{(p + 1)!} \left( \frac{j_0 + 1}{j_0 + p + 2} \tilde{x}_{j_0} \right) = \frac{\Pi_{\ell=1}(j_0 + \ell)}{(p + 1)!} \left( y_{j_0} - \frac{p + 1}{j_0 + p + 2} z_{j_0} \right) = \frac{\Pi_{\ell=1}(j_0 + \ell)}{(p + 1)!} y_{j_0} - \frac{\Pi_{\ell=1}(j_0 + \ell)}{p!} z_{j_0}.
\]
Combining (20), (21) and (22) yields
\[
\psi_{j_0+1}(z_{j_0+1}, \tau_{j_0+1}) \\
\geq \min_{z \in \mathbb{R}^d} \left\{ \frac{\Pi_{\ell=2}^{p+1}(j_0 + \ell)}{p!} (z - z_{j_0})^\top (\nabla f(\bar{x}_{j_0+1}) + \bar{\xi}_{j_0+1}) + \frac{\tau_{j_0} \|z - z_{j_0}\|^{p+1}}{2(p+1)} \right\} \\
+ \frac{\Pi_{\ell=1}^{p+1}(j_0 + 1 + \ell)}{(p+1)!} (y_{j_0} - \bar{x}_{j_0+1})^\top (\nabla f(\bar{x}_{j_0+1}) + \bar{\xi}_{j_0+1}) + \frac{\Pi_{\ell=1}^{p+1}(j_0 + 1 + \ell)}{(p+1)!} F(\bar{x}_{j_0+1}).
\]

Furthermore, since \( j_0 \) is a successful iteration, we have
\[
(y_{j_0} - \bar{x}_{j_0+1})^\top (\nabla f(\bar{x}_{j_0+1}) + \bar{\xi}_{j_0+1}) \geq \eta \|y_{j_0} - \bar{x}_{j_0+1}\|^{p+1}
\]
by Lemma A.5
\[
\geq \eta \left( \frac{\|\nabla f(\bar{x}_{j_0+1}) + \bar{\xi}_{j_0+1}\|}{\rho_p + \sigma_2 + \kappa_\theta} \right)^{1+\frac{1}{p}}.
\]

Thus, it suffices to establish
\[
\eta \Pi_{\ell=1}^{p+1}(j_0 + 1 + \ell) \left( \frac{\|\nabla f(\bar{x}_{j_0+1}) + \bar{\xi}_{j_0+1}\|}{\rho_p + \sigma_2 + \kappa_\theta} \right)^{1+\frac{1}{p}} + \frac{\tau_{j_0} \|z - z_{j_0}\|^{p+1}}{2(p+1)} \\
+ \frac{\Pi_{\ell=1}^{p+1}(j_0 + 1 + \ell)}{p!} (z - z_{j_0})^\top (\nabla f(\bar{x}_{j_0+1}) + \bar{\xi}_{j_0+1}) \geq 0, \forall z \in \mathbb{R}^d.
\]

Indeed, applying (17) with
\[
f = \frac{\Pi_{\ell=2}^{p+1}(j_0 + \ell)}{p!} (\nabla f(\bar{x}_{j_0+1}) + \bar{\xi}_{j_0+1}), \quad s = z - z_{j_0}, \quad \sigma = \frac{\tau_{j_0}}{2}, \quad q = p + 1
\]
we obtain that
\[
\frac{\Pi_{\ell=2}^{p+1}(j_0 + \ell)}{p!} (z - z_{j_0})^\top (\nabla f(\bar{x}_{j_0+1}) + \bar{\xi}_{j_0+1}) + \frac{\tau_{j_0} \|z - z_{j_0}\|^{p+1}}{2(p+1)} \\
\geq - \frac{p}{p+1} \left( \frac{2}{\tau_{j_0}} \right)^{\frac{1}{p}} \left( \frac{\Pi_{\ell=2}^{p+1}(j_0 + \ell)}{p!} \|\nabla f(\bar{x}_{j_0+1}) + \bar{\xi}_{j_0+1}\| \right)^{1+\frac{1}{p}}.
\]

Therefore, (23) is equivalent to
\[
\tau_{j_0} \geq \frac{2(\rho_p + \sigma_2 + \kappa_\theta)^{p+1}}{\eta^p} \left( \frac{p}{p+1} \right)^p \left( \frac{\Pi_{\ell=2}^{p+1}(j_0 + \ell)}{p!} \right)^p \left( \frac{(p+1)!}{\Pi_{\ell=1}^{p+1}(j_0 + 1 + \ell)} \right)^p \\
= \frac{2(\rho_p + \sigma_2 + \kappa_\theta)^{p+1}}{\eta^p} \left( \frac{\Pi_{\ell=2}^{p+1}(j_0 + \ell)}{(j_0 + p + 2)^p} \right)^p \frac{p^p}{p!}.
\]

Finally, observe that the conclusion would follow if
\[
\tau_{j_0} \geq \frac{2(\rho_p + \sigma_2 + \kappa_\theta)^{p+1}}{\eta^p} \frac{p^{p-1}}{(p-1)!}.
\]
Furthermore, it suffices to show the inequality on the right hand side. Denote \(x_0 \in \mathbb{R}^d\) to be the initial iterate of SAS and \(\bar{x}_0^m \in \mathbb{R}^d\) to be the global minimizer of \(m(x; x_0, \sigma_i)\) over \(\mathbb{R}^d\). Since \(\bar{x}_0\) is also the output returned by SAS, it holds that \(m(\bar{x}_0^m; x_0, \sigma_i) \leq m(\bar{x}_0; x_0, \sigma_i) \leq m(x_0; x_0, \sigma_i) = F(x_0)\). Moreover, we have proved in Lemma A.7 that \(\sigma_i \geq \sigma_{\text{min}}\) for all \(\sigma_i\) in SAS, which implies that \(m(x; x_0, \sigma_{\text{min}}) \leq m(x; x_0, \sigma_i)\) for all \(x\) and thus \(L(x_0, \sigma_i) \leq L(x_0, \sigma_{\text{min}})\). Then according to (11),

\[
\|x_0^m - x^*\| \leq D \quad \text{and} \quad \|\bar{x}_0 - x^*\| \leq D.
\]

Besides

\[
F(\bar{x}_0) \leq m(\bar{x}_0; x_0, \sigma_i) = m(\bar{x}_0; x_0, \sigma_i) - m(\bar{x}_0^m; x_0, \sigma_i) + m(\bar{x}_0^m; x_0, \sigma_i).
\]

Furthermore, \(R(z)\) is convex because

\[
\nabla^2 R(z) = \frac{1}{2} \left( p \|z - x_0\|^{p-1} (z - x_0)^\top + \|z - x_0\|^{p-1} I \right) \succeq 0,
\]

which implies that \(m(x; x_0, \sigma_i)\) is convex as \(\mathcal{m}(x; x_0)\) is convex. Therefore, there exists some \(\tilde{\xi}_0 \in \partial r(\bar{x}_0)\) such that

\[
m(\bar{x}_0; x_0, \sigma_i) - m(\bar{x}_0^m; x_0, \sigma_i) \leq - \langle \nabla m(\bar{x}_0; x_0) + \sigma_i \|\bar{x}_0 - x_0\|^{p-1} (\bar{x}_0 - x_0) + \tilde{\xi}_0 \rangle (\bar{x}_0^m - \bar{x}_0)
\]

\[
\leq \|\nabla m(\bar{x}_0; x_0) + \sigma_i \|\bar{x}_0 - x_0\|^{p-1} (\bar{x}_0 - x_0) + \tilde{\xi}_0 \|\bar{x}_0 - \bar{x}_0^m\|
\]

\[
\leq \kappa_\theta \|\bar{x}_0 - x_0\|^{p} \|\bar{x}_0 - \bar{x}_0^m\|.
\]

On the other hand,

\[
m(\bar{x}_0^m; x_0, \sigma_i) = \mathcal{m}(\bar{x}_0^m; x_0) + \sigma_i \|\bar{x}_0^m - x_0\|^{p+1} \|\bar{x}_0^m\| + r(\bar{x}_0^m)
\]

\[
\leq \mathcal{m}(z; x_0) + \sigma_i \|z - x_0\|^{p+1} \|z\| + r(z)
\]

\[
\leq f(z) + \kappa_p \|z - x_0\|^{p+1} + \bar{\kappa}_p \|z - x_0\|^{p} + \sigma_i \|z - x_0\|^{p+1} \|z\| + r(z)
\]

\[
= F(z) + \frac{(p + 1)\kappa_p + \bar{\kappa}_p}{p + 1} \|z - x_0\|^{p+1} + \bar{\kappa}_p \|z - x_0\|^{p}.
\]
Combining the above two inequalities, we have

\[
\psi_0(z, \tau_0) = F(\bar{x}_0) + \frac{\tau_0 \|z - \bar{x}_0\|^{p+1}}{2(p+1)} \leq \left(m(\bar{x}_0; x_0, \sigma_i) + m(\bar{x}_0^m; x_0, \sigma_i) + \frac{\tau_0 \|z - \bar{x}_0\|^{p+1}}{2(p+1)} \right)
\]

which the last inequality is due to (24) and (11).

\[\square\]

Next, we proceed to analyzing all the iterates in AAS.

**Theorem A.8** The sequence \(\{\bar{x}_j, j \geq 0\}\) generated by AAS in UAA satisfies

\[
\frac{\Pi_{\ell=1}^{p+1}(j + \ell)}{(p+1)!} F(\bar{x}_j) \leq \psi_j(z, \tau_j) \leq \psi_j(z, \tau_{j+1}) \leq F(z) + \frac{(p+1)\kappa_p + \bar{\sigma}_1}{p+1} \|z - x_0\|^{p+1} + \frac{\tau_0 \|z - \bar{x}_0\|^{p+1}}{2(p+1)} + \kappa_\theta(2D)^{p+1},
\]

which the last inequality is due to (24) and (11).

Proof. By the way in which \(\psi_{j+1}(z_{j+1}, \tau_{j+1})\) is updated in AAS, we have

\[
\frac{\Pi_{\ell=1}^{p+1}(j + 1 + \ell)}{(p+1)!} F(\bar{x}_{j+1}) \leq \psi_j(z_{j+1}, \tau_{j+1}) \leq \psi_j(z_j, \tau_{j+1}) \leq \psi_j(z_j, \tau_{j+1}), \quad \forall j \geq 0.
\]

It thus suffices to show the inequality on the right hand side by induction. The base case of \(j = 0\) has already been proved in Theorem A.7. We now assume the result holds for some \(j = j_0\). For the case
Then, taking Proof of Theorem 3.2:

Finally, we are in position to prove Theorem 3.2. This completes the proof.

\[ \psi_{j_0+1}(\bar{x}_{j_0+1}, \bar{\tau}_{j_0+1}) \leq \psi_{j_0+1}(z, \tau_{j_0+1}) \]

\[ = l_{j_0}(z) + \Delta l_{j_0}(z; \bar{x}_{j_0+1}, \bar{\tau}_{j_0+1}) + \frac{\tau_{j_0+1} \| z - x_0 \|^{p+1}}{2(p+1)} \]

\[ \leq \frac{\prod_{\ell=1}^{j_0+1}(j_0 + \ell)}{(p+1)!} F(z) + \frac{(p+1)k_p + s_1}{p+1} \| z - x_0 \|^{p+1} \]

\[ + \kappa_0(2D)^{p+1} + \frac{\prod_{\ell=2}^{j_0+1}(j_0 + \ell)}{p!} \left[ F(\bar{x}_{j_0+1}) + (z - \bar{x}_{j_0+1})^\top (\nabla g(\bar{x}_{j_0+1}) + \bar{\tau}_{j_0+1}) \right] \]

\[ \overset{\text{Assumption A.1}}{\leq} \frac{\prod_{\ell=1}^{j_0+1}(j_0 + 1 + \ell)}{(p+1)!} F(z) + \frac{(p+1)k_p + s_1}{p+1} \| z - x_0 \|^{p+1} \]

\[ + \kappa_0(2D)^{p+1} + \frac{\prod_{\ell=2}^{j_0+1}(j_0 + 1 + \ell)}{p!} \]

\[ \leq \frac{\prod_{\ell=1}^{j_0+1}(j_0 + 1 + \ell)}{(p+1)!} \]

\[ \leq \frac{\prod_{\ell=1}^{j_0+1}(j_0 + 1 + \ell)}{(p+1)!} \]

where the second last inequality is due to the mathematical induction and \( \tau_j \) is monotonically increasing. This completes the proof.

Finally, we are in position to prove Theorem 3.2.

Proof of Theorem 3.2: Recall that in the proof of Theorem A.7 we have shown \( \| x^* - \bar{x}_0 \| \leq D \). Then, taking \( z = x^* \) in (25) yields that

\[ \frac{\prod_{\ell=1}^{j_0+1}(j_0 + \ell)}{(p+1)!} F(\bar{x}_j) \leq \frac{\prod_{\ell=1}^{j_0+1}(j_0 + \ell)}{(p+1)!} F(x^*) + \frac{(p+1)k_p + s_1}{p+1} \| x^* - x_0 \|^{p+1} \]

\[ + \frac{\tau_0 \| x^* - \bar{x}_0 \|^{p+1}}{2(p+1)} + \kappa_0(2D)^{p+1} \]

\[ \leq \frac{\prod_{\ell=1}^{j_0+1}(j_0 + 1 + \ell)}{(p+1)!} \]

which further implies that

\[ F(\bar{x}_j) - F(x^*) \leq \frac{(p+1)! \left( \frac{2(p+1)k_p + 2s_1 + \tau_0}{2(p+1)} D^{p+1} + \kappa_p D^p + \kappa_0(2D)^{p+1} \right)}{\prod_{\ell=1}^{j_0+1}(j_0 + \ell)} \]

Finally, combining the above inequality with Lemmas A.1, A.2 and A.6 the conclusion follows.