Krylov Cubic Regularized Newton: A Subspace Second-Order Method with Dimension-Free Convergence Rate

Ruichen Jiang
UT Austin

Parameswaran Raman
Amazon Web Services

Shoham Sabach
Technion

Aryan Mokhtari
UT Austin

Mingyi Hong
University of Minnesota

Volkan Cevher
EPFL

Abstract

Second-order optimization methods, such as cubic regularized Newton methods, are known for their rapid convergence rates; nevertheless, they become impractical in high-dimensional problems due to their substantial memory requirements and computational costs. One promising approach is to execute second-order updates within a lower-dimensional subspace, giving rise to *subspace second-order* methods. However, the majority of existing subspace second-order methods randomly select subspaces, consequently resulting in slower convergence rates depending on the problem's dimension d. In this paper, we introduce a novel subspace cubic regularized Newton method that achieves a dimension-independent global convergence rate of $\mathcal{O}\left(\frac{1}{mk}+\frac{1}{k^2}\right)$ for solving convex optimization problems. Here, m represents the subspace dimension, which can be significantly smaller than d. Instead of adopting a random subspace, our primary innovation involves performing the cubic regularized Newton update within the *Krylov subspace* associated with the Hessian and the gradient of the objective function. This result marks the first instance of a dimension-independent convergence rate for a subspace second-order method. Furthermore, when specific spectral conditions of the Hessian are met, our method recovers the convergence rate of a full-dimensional cubic regularized Newton method. Numerical experiments show our method converges faster than existing random subspace methods, especially for high-dimensional problems.

1 INTRODUCTION

In this paper, we consider the following unconstrained minimization problem

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

where $f:\mathbb{R}^d\to\mathbb{R}$ is convex and twice continuously differentiable. We focus on the use of second-order methods for solving this problem, particularly the Cubic Regularized Newton (CRN) method (Griewank, 1981; Nesterov and Polyak, 2006). The CRN method stands out for its fast convergence rate. Specifically, when dealing with convex functions, it exhibits a global convergence rate of $\mathcal{O}(1/k^2)$, with k denoting the number of iterations. Furthermore, in cases where f is strongly convex, it attains a superlinear convergence rate (Nesterov and Polyak, 2006). Nevertheless, the main drawback associated with the CRN method is its substantial computational cost per iteration, particularly when the problem's dimensionality d is high, leading to unfavorable scaling. This arises from the necessity to solve a cubic subproblem at each iteration, demanding a minimum of $\mathcal{O}(d^3)$ arithmetic operations. As a result, CRN becomes impractical for optimization problems with high dimensions, a common scenario in modern machine learning applications.

To reduce the computational cost, Hanzely et al. (2020) proposed the stochastic subspace cubic Newton (SSCN) method, which can be regarded as a subspace variant of CRN. Inspired by first-order coordinate descent methods (Luo and Tseng, 1992; Nesterov, 2012; Wright, 2015), their key proposition is to solve

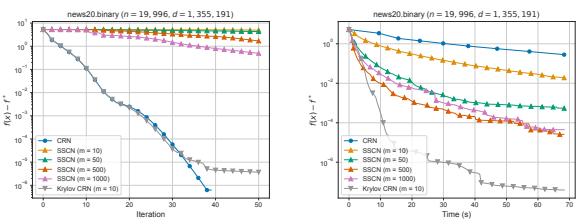
Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

Table 1: The comparison of CRN, SSCN, and our method in terms of per-iteration cost and convergence rate. *We assume that the cost of computing the subspace gradient and the subspace Hessian is $\mathcal{O}(m)$ and $\mathcal{O}(m^2)$, respectively. **We assume that the cost of Hessian-vector product evaluations is $\mathcal{O}(d)$.

Methods	Per-iteration cost	Convergence rate
CRN (Nesterov and Polyak, 2006)	$\mathcal{O}(d^3)$	$\mathcal{O}(\frac{1}{k^2})$
SSCN (Hanzely et al., 2020)	$\mathcal{O}(m^3)^*$	$\mathcal{O}(\frac{d-m}{m}\cdot\frac{1}{k}+\frac{d^2}{m^2}\cdot\frac{1}{k^2})$
Krylov CRN (ours)	$\mathcal{O}(md)^{**}$	$\mathcal{O}(\frac{1}{mk}+\frac{1}{k^2})$

the cubic subproblem over a random low-dimensional subspace of dimension $m \ll d$, instead of the full-dimensional space $\mathbb{R}^d$. As a result, this strategy effectively reduces the dimension of the cubic subproblem to m, and thus it can be solved efficiently using $\mathcal{O}(m^3)$ arithmetic operations. However, this efficiency comes at a cost: SSCN suffers from a slower convergence rate of $\mathcal{O}\left(\frac{d-m}{m}\cdot\frac{1}{k}+\left(\frac{d}{m}\right)^2\cdot\frac{1}{k^2}\right)$ that scales with the problem's dimension d. Additionally, formulating the low-dimensional cubic subproblem requires computing the gradient and the Hessian of the objective over the chosen subspace, which also needs to be taken into consideration. In certain special cases, such as generalized linear models, they can be computed with a complexity of $\mathcal{O}(m)$ and $\mathcal{O}(m^2)$, respectively, as demonstrated by (Hanzely et al., 2020). In this case, the dominant cost comes from solving the cubic subproblem, leading to an overall complexity of $\mathcal{O}(m^3)$. In Table 1, we report the per-iteration cost of SSCN for such favorable cases. In general, however, the cost of computing the subspace gradient and the subspace Hessian could be dependent on the problem's dimension d. For instance, Gower et al. (2019) proposed computing the subspace Hessian via m back-propagation passes. Since each back-propagation requires $\mathcal{O}(d)$ arithmetic operations, in this case the per-iteration cost could be $\mathcal{O}(md)$.

Given the discussions above, we are motivated by the following question: "*Can we improve the dimensional dependence of subspace second-order methods?*" Intuitively, this problem stems from the fact that the subspace is chosen uniformly random at each iteration, oblivious to the objective function we aim to optimize. As a result, such a random subspace is unlikely to contain a "good" descent direction of the objective, hindering the convergence of the subspace method. In this paper, we argue that employing a subspace customized to the local geometry of the objective yields a convergence rate that is independent of the dimensionality. Specifically, we propose the Krylov CRN method, where we perform the CRN update over the



(a) (b)

Figure 1: Numerical comparisons of CRN, SSCN, and our proposed Krylov CRN method on a logistic regression problem. Here, m denotes the dimension of the subspace, n is the number of samples, and d is the number of parameters. See Section 5 for full details.

Krylov subspace associated with the Hessian and the gradient of the objective function. Our contributions are summarized as follows:

- In the convex case, we prove a dimension-free convergence rate of $\mathcal{O}(\frac{1}{mk}+\frac{1}{k^2})$ for our proposed method, where m is the subspace dimension and k is the number of iterations. In particular, we shave a factor of $\mathcal{O}(d)$ from the iteration complexity of SSCN when $m \ll d$ (see Table 1). Additionally, in the strongly convex case, our proposed method achieves a linear rate of convergence, and we again shave a $\mathcal{O}(d)$ factor from the iteration complexity of SSCN when $m \ll d$. Furthermore, we show that our method can be implemented using the Lanczos method, requiring one gradient evaluation and m Hessian-vector products per iteration, and the resulting cubic subproblem can be solved using $\mathcal{O}(m)$ arithmetic operations. Hence, in the worst case, the per-iteration cost of our method is $\mathcal{O}(md)$, resulting from the computation of m Hessian-vector products.

- We show that our method is capable of exploiting the spectral structure of the objective's Hessian. Specifically, we characterize the convergence rate of our method in terms of a spectral quantity of the objective's Hessian (cf. Theorem 1), which could lead to faster convergence rates when the Hessian possesses some structure. For instance, if the Hessian has at most m distinct eigenvalues, then our method recovers the $\mathcal{O}(1/k^2)$ rate of the full-dimensional CRN method.

- To demonstrate the efficacy of our algorithm, we perform numerical experiments on high-dimensional logistic regression problems, as exemplified by Figure 1. Specifically, we observe from Figure (a) that our proposed Krylov CRN method makes much more progress than SSCN in

each iteration and closely follows the loss curve of the full-dimensional CRN, even with a modest subspace dimension of $m = 10$. Moreover, given the same amount of computation time, Figure (b) shows that our proposed method can converge much faster than both SSCN and the full-dimensional CRN.

Additional related work. The idea of using the Krylov subspace method for solving the subproblem in CRN has been previously explored by Cartis et al. (2011) and Carmon and Duchi (2018). However, there are two key distinctions between their work and ours. Firstly, they focus on solving nonconvex optimization problems and establish results for finding approximate stationary points, whereas our focus is on the convex and strongly convex settings. Secondly, and more importantly, their analysis requires the Krylov subspace solution to be an approximate minimizer of the full-dimensional cubic subproblem, and hence the subspace dimension m needs to depend on the final target accuracy ϵ. Consequently, to attain a final accuracy of ϵ, they require the subspace dimension to be $m = \tilde{\mathcal{O}}(\epsilon^{-\frac{1}{4}})$ (Carmon and Duchi, 2020). In contrast, our results hold for any constant value of m, which can be independent of the iteration index k or the target accuracy ϵ.

In addition to CRN, another classical second-order method is the damped Newton's method, and its subspace variants have also been considered by Gower et al. (2019); Hanzely (2023). Since their convergence rates are shown under a different set of assumptions, their results are not directly comparable with ours. In addition, we note that their convergence rates suffer from the same issue of dimensional dependence as they also rely on random subspaces.

2 PRELIMINARIES

In this section, we first introduce the necessary assumptions and highlight the auxiliary results arising from these conditions. Then, in Section 2.1, we discuss the original cubic regularized Newton (CRN) method as well as its subspace variants studied in the literature. Note that our required assumptions are all common in the analysis of CRN-type methods (Nesterov and Polyak, 2006).

Assumption 1. *The function $f : \mathbb{R}^d \to \mathbb{R}$ is convex.*

Assumption 2. *The function f is bounded from below and has bounded level-sets.*

Assumption 3. *The Hessian $\nabla^2 f$ is L_2-Lipschitz: $\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \leq L_2\|\mathbf{x} - \mathbf{y}\|$, for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.*

Utilizing conventional arguments (see, e.g., (Nesterov, 2018, Lemma 1.2.4)), one key implication of Assumption 3 is that we can control the difference between the function $f(\mathbf{x})$ and its quadratic approximation, as demonstrated in the following proposition.

Proposition 1. *If Assumption 3 holds, then for any $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$, we have $|f(\mathbf{x}) - f(\mathbf{x}') - \nabla f(\mathbf{x}')^\top(\mathbf{x} - \mathbf{x}') - \frac{1}{2}(\mathbf{x} - \mathbf{x}')^\top \nabla^2 f(\mathbf{x}')(\mathbf{x} - \mathbf{x}')| \leq \frac{L_2}{6}\|\mathbf{x} - \mathbf{x}'\|^3$.*

2.1 Subspace Cubic Regularized Newton

In this part, we start by providing an overview of the classic full-dimensional CRN method (Nesterov and Polyak, 2006) as well as its subspace variant. Recall that the CRN method is motivated by the following simple observation: using Proposition 1, we can upper bound the function f by its quadratic approximation with a cubic regularizer. Specifically, let $\mathbf{g}_k$ and $\mathbf{H}_k$ denote the vector $\nabla f(\mathbf{x}_k)$ and the matrix $\nabla^2 f(\mathbf{x}_k)$, respectively; we will use these notations throughout the paper. Then, given a regularization parameter $M \geq L_2$ and the current iterate $\mathbf{x}_k$, we have

$$f(\mathbf{x}) \leq f(\mathbf{x}_k) + \mathbf{g}_k^\top(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^\top \mathbf{H}_k(\mathbf{x} - \mathbf{x}_k) + \frac{M}{6}\|\mathbf{x} - \mathbf{x}_k\|^3. \tag{1}$$

Thus, a reasonable choice is to select the new iterate $\mathbf{x}_{k+1}$ as the minimizer of the upper bound of $f(\mathbf{x})$ that is given in the right-hand side of (1). Specifically, starting with any $\mathbf{x}_0 \in \mathbb{R}^d$, the update rule of CRN can be written as

$$\mathbf{s}_k := \arg\min_{\mathbf{s} \in \mathbb{R}^d} \Big\{ \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M}{6}\|\mathbf{s}\|^3 \Big\} \tag{2}$$
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k.$$

It is well-known that the CRN method achieves a fast convergence rate of $\mathcal{O}(1/k^2)$ if Assumptions 1-3 hold (Nesterov and Polyak, 2006). However, solving the cubic subproblem in (2) could be computationally prohibitive, particularly in a high-dimensional setting where d is large. To better elaborate on this issue, note that the standard approach for solving the above subproblem (as discussed in (Conn et al., 2000; Cartis et al., 2011)) is to reformulate (2) as a one-dimensional nonlinear equation in λ, which is given by $\lambda = \frac{M}{2}\|(\mathbf{H}_k + \lambda \mathbf{I})^{-1}\mathbf{g}_k\|$. By applying Newton's method to find its root, to be denoted by λ^*, the solution of (2) can then be determined as $\mathbf{s}_k = -(\mathbf{H}_k + \lambda^* \mathbf{I})^{-1}\mathbf{g}_k$. Therefore, finding $\mathbf{s}_k$ at each iteration demands solving multiple linear systems of equations of dimension d, which requires $\mathcal{O}(d^3)$ arithmetic operations. Hence, the cost of finding the solution of (2) in CRN scales poorly with the dimension d.

To mitigate this issue, subspace variants of the CRN method have been proposed in the literature

by Doikov and Richtárik (2018) and Hanzely et al. (2020). Their main idea is to restrict the variable $\mathbf{s}$ in the subproblem in (2) to a low-dimensional subspace $\mathcal{V}_k$, thus reducing the effective dimension of the subproblem. Concretely, let $\mathbf{V}_k \in \mathbb{R}^{d\times m}$ be a tall matrix whose columns form an orthogonal basis for $\mathcal{V}_k$, where m is the subspace dimension. Then, solving the cubic subproblem in (2) over the subspace $\mathcal{V}_k$ is equivalent to

$$\begin{aligned}\mathbf{z}_k &:= \arg\min_{\mathbf{z}\in\mathbb{R}^m}\Big\{\tilde{\mathbf{g}}_k^\top \mathbf{z} + \frac{1}{2}\mathbf{z}^\top \tilde{\mathbf{H}}_k \mathbf{z} + \frac{M}{6}\|\mathbf{z}\|^3\Big\}, \qquad (3)\\ \mathbf{s}_k &:= \mathbf{V}_k \mathbf{z}_k,\end{aligned}$$

where $\tilde{\mathbf{g}}_k := \mathbf{V}_k^\top \mathbf{g}_k \in \mathbb{R}^m$ and $\tilde{\mathbf{H}}_k := \mathbf{V}_k^\top \mathbf{H}_k \mathbf{V}_k \in \mathbb{R}^{m\times m}$ are the gradient and Hessian in the subspace, respectively. Note that the effective dimension of the subproblem in (3) is m and hence it can be solved in $O(m^3)$ time. In particular, Doikov and Richtárik (2018) considered objective functions with a block separable structure, and their method defines the subspace $\mathcal{V}_k$ and its corresponding projection matrix $\mathbf{V}_k$ by sampling a random block of coordinates at each iteration. Later, Hanzely et al. (2020) extended this idea of the subspace CRN method to general objective functions and general random subspaces satisfying the assumption $\mathbb{E}[\mathbf{V}_k \mathbf{V}_k^\top] = \frac{m}{d}\mathbf{I}$ for all $k \geq 0$.

Although the proposed randomized subspace CRN methods in (Doikov and Richtárik, 2018; Hanzely et al., 2020) successfully reduce the cost of solving the subproblem in the CRN method, their convergence rate depends on the problem's dimension d, as highlighted in Table 1. Moreover, in addition to Assumptions 1-3, they also require an extra assumption that the objective function Hessian has bounded eigenvalues, i.e., $\nabla^2 f(\mathbf{x}) \preceq L_1 \mathbf{I}$ for all $\mathbf{x} \in \mathbb{R}^d$. Under these assumptions, SSCN is shown to achieve a rate of $\mathcal{O}(\frac{d-m}{m}\cdot\frac{L_1}{k} + \frac{d^2}{m^2}\cdot\frac{1}{k^2})$. This issue raises the question of whether there is a better choice of the subspace so that the convergence rate of the subspace CRN method is independent of the dimension and has a better dependence on L_1, while ensuring that the cost of solving the subproblem remains affordable. In the next section, we show that our proposed Krylov subspace CRN method can achieve this goal.

3 THE PROPOSED ALGORITHM

In this section, we first explain the rationale behind the use of the Krylov subspace before introducing our proposed Krylov CRN method.

3.1 Rationale behind the Krylov Subspace

To motivate the use of the Krylov subspace, let us examine how introducing the subspace $\mathcal{V}_k$ would change the convergence analysis and under what conditions on $\mathcal{V}_k$ the subspace method can still retain the fast convergence rate of $\mathcal{O}(1/k^2)$. In the analysis of the CRN method in (Nesterov and Polyak, 2006), given the current iterate $\mathbf{x}_k$ and the next iterate $\mathbf{x}_{k+1}$ which is computed based on (2), we use the following crucial inequality

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M}{6}\|\mathbf{s}\|^3, \quad (4)$$

which is true for any $M \geq L_2$ and for all $\mathbf{s} \in \mathbb{R}^d$. Importantly, (4) provides an upper bound on $f(\mathbf{x}_{k+1})$ for *any* $\mathbf{s} \in \mathbb{R}^d$, which gives us the freedom to choose a well-designed $\mathbf{s}$ related to the optimal solution $\mathbf{x}^*$ in the convergence analysis.

Given that the inequality in (4) plays a significant role in demonstrating the $\mathcal{O}(1/k^2)$ convergence rate of CRN, we need to investigate the equivalent inequality that can be established for the subspace variant of the CRN method. This inequality may assist us in making well-informed choices for the subspace $\mathcal{V}_k$ to minimize its impact on the convergence rate. More precisely, if we follow the update in (3), then we can show that for any $\mathbf{s} \in \mathbb{R}^d$, we have

$$\begin{aligned}f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{P}_k \mathbf{s} + \frac{1}{2}(\mathbf{P}_k \mathbf{s})^\top \mathbf{H}_k \mathbf{P}_k \mathbf{s}\\ + \frac{M}{6}\|\mathbf{P}_k \mathbf{s}\|^3, \qquad (5)\end{aligned}$$

where $\mathbf{P}_k := \mathbf{V}_k \mathbf{V}_k^\top$ is the orthogonal projection matrix associated to the subspace $\mathcal{V}_k$ (for more details check the proof of Lemma 10 in the Appendix). By comparing (5) and (4) term by term, we observe that (4) still holds true if $\mathbf{P}_k$ is selected such that the following conditions hold for all $\mathbf{s} \in \mathbb{R}^d$:

(A) $\mathbf{g}_k^\top \mathbf{P}_k \mathbf{s} \leq \mathbf{g}_k^\top \mathbf{s}$.

(B) $(\mathbf{P}_k \mathbf{s})^\top \mathbf{H}_k \mathbf{P}_k \mathbf{s} \leq \mathbf{s}^\top \mathbf{H}_k \mathbf{s}$.

(C) $\|\mathbf{P}_k \mathbf{s}\| \leq \|\mathbf{s}\|$.

Note that Condition (C) holds automatically since $\mathbf{P}_k$ is an orthogonal projection matrix. On the other hand, the first two conditions impose restrictions on the matrix $\mathbf{P}_k$, which in turn constrains the subspace $\mathcal{V}_k$ that we can choose. Below, we will outline the conditions under which the requirements in (A) and (B) would be met.

Proposition 2. *Let $\mathbf{P}_k$ be the orthogonal projection matrix associated to the subspace $\mathcal{V}_k$.*

(i) Condition (A) holds true if and only if $\mathbf{g}_k \in \mathcal{V}_k$.

(ii) Condition (B) holds true if and only if $\mathcal{V}_k$ is an invariant subspace with respect to the matrix $\mathbf{H}_k$, i.e., $\mathbf{H}_k \mathbf{s} \in \mathcal{V}_k$ for any $\mathbf{s} \in \mathcal{V}_k$.

Algorithm 1 Krylov cubic regularized Newton

Input: Initial point $\mathbf{x}_0 \in \mathbb{R}^d$, subspace dimension m, regularization parameter $M > 0$
for $k = 0, 1, \ldots,$ **do**
  Set $(\mathbf{V}_k, \tilde{\mathbf{g}}_k, \tilde{\mathbf{H}}_k) = \textsc{Lanczos}(\mathbf{H}_k, \mathbf{g}_k; m)$
  Solve the cubic subproblem

$$\mathbf{z}_k = \arg\min_{\mathbf{z} \in \mathbb{R}^m} \Big\{ \tilde{\mathbf{g}}_k^\top \mathbf{z} + \frac{1}{2} \mathbf{z}^\top \tilde{\mathbf{H}}_k \mathbf{z} + \frac{M}{6} \|\mathbf{z}\|^3 \Big\},$$

  Update $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{V}_k \mathbf{z}_k$
end for

From Proposition 2, we immediately obtain that $\mathbf{H}_k \mathbf{g}_k \in \mathcal{V}_k$. Moreover, by repeatedly applying (ii), it follows from induction that $\mathbf{H}_k^i \mathbf{g}_k \in \mathcal{V}_k$ for any $i \geq 0$. Thus, we conclude that the minimal subspace satisfying both conditions (A) and (B) is given by the linear span of $\{\mathbf{H}_k^i \mathbf{g}_k\}_{i=0}^{\infty}$, which is exactly the maximal *Krylov subspace* generated by $\mathbf{H}_k$ and $\mathbf{g}_k$. Formally, the j-th Krylov subspace generated by a symmetric matrix $\mathbf{A}$ and a vector $\mathbf{b}$ is defined as

$$\mathcal{K}_j(\mathbf{A}, \mathbf{b}) = \text{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \ldots, \mathbf{A}^{j-1}\mathbf{b}\}.$$

Moreover, it can be shown that (Saad, 2011, Chapter 6) there exists an integer r_0 such that $\mathcal{K}_j(\mathbf{A}, \mathbf{b}) = \mathcal{K}_{r_0}(\mathbf{A}, \mathbf{b})$ for all $j \geq r_0$, and we call $\mathcal{K}_{r_0}(\mathbf{A}, \mathbf{b})$ as the maximal Krylov subspace. In this case, the dimension of the maximal Krylov subspace is r_0.

Now given these definitions, if we select $\mathcal{V}_k$ in (3) as the maximal Krylov subspace generated by $\mathbf{H}_k$ and $\mathbf{g}_k$ denoted by $\mathcal{K}_{r_0}(\mathbf{H}_k, \mathbf{g}_k)$, then the key inequality in (4) remains valid and we retain the same convergence rate as the full-dimensional CRN method. However, the dimension of the maximal Krylov subspace can be as large as d, which contradicts our goal of dimension reduction. To solve this problem, we propose using the Krylov subspace up to dimension m, which is $\mathcal{K}_m(\mathbf{H}_k, \mathbf{g}_k)$. In this case, Condition (A) still holds exactly since $\mathbf{g}_k \in \mathcal{K}_m(\mathbf{H}_k, \mathbf{g}_k)$ for any $m \geq 1$, and the only violated condition due to this approximation is Condition (B). In fact, as we shall show in Section 4, the approximation error corresponding to Condition (B) would be independent of the problem's dimension d. In comparison, when using random subspace selection as in SSCN, both conditions in (A) and (B) only hold approximately and the induced approximation error depends on the ratio m/d, resulting in a convergence rate that inevitably depends on d.

3.2 Krylov Cubic Regularized Newton

Next, we present our proposed Krylov CRN method, which is a particular instance of the subspace CRN method in (3). Specifically, we choose the subspace $\mathcal{V}_k$ to be the m-th Krylov subspace $\mathcal{K}_m(\mathbf{H}_k, \mathbf{g}_k)$ and let $\mathbf{V}_k \in \mathbb{R}^{d \times m}$ be the matrix formed by its orthonormal basis. Our method is summarized in Algorithm 1.

Algorithm 2 $(\mathbf{V}, \tilde{\mathbf{b}}, \tilde{\mathbf{A}}) = \textsc{Lanczos}(\mathbf{A}, \mathbf{b}; m)$

Input: $\mathbf{A} \in \mathbb{R}^{d \times d}$, $\mathbf{b} \in \mathbb{R}^d$, and the dimension m
Initialize: $\mathbf{v}_1 = \mathbf{b}/\|\mathbf{b}\|$, $\beta_1 = 0$, $\mathbf{v}_0 = 0$
for $j = 1, 2, \ldots, m$ **do**
  $\mathbf{w}_j \leftarrow \mathbf{A}\mathbf{v}_j - \beta_j \mathbf{v}_{j-1}$
  $\alpha_j \leftarrow \mathbf{w}_j^\top \mathbf{v}_j$
  $\mathbf{w}_j \leftarrow \mathbf{w}_j - \alpha_j \mathbf{v}_j$
  $\beta_{j+1} \leftarrow \|\mathbf{w}_j\|_2$
  $\mathbf{v}_{j+1} \leftarrow \mathbf{w}_j / \beta_{j+1}$
end for
Output: $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$, $\tilde{\mathbf{b}} = \|\mathbf{b}\|\mathbf{e}_1$, and $\tilde{\mathbf{A}} = \text{tridiag}(\{\beta_j\}_{j=2}^m, \{\alpha_j\}_{j=1}^m, \{\beta_j\}_{j=2}^m)$

The only missing piece in Algorithm 1 is how to compute the orthonormal basis of $\mathcal{V}_k$ to form the matrix $\mathbf{V}_k$, as well as the subspace gradient $\tilde{\mathbf{g}}_k$ and the subspace Hessian $\tilde{\mathbf{H}}_k$. To achieve this goal, we use the Lanczos method as shown in Algorithm 2, which is a commonly used method for computing an orthonormal basis for the Krylov subspace (Lanczos, 1950). Specifically, given an input matrix $\mathbf{A}$, an input vector $\mathbf{b}$ and the target dimension m, Algorithm 2 iteratively generates a sequence of orthonormal vectors $\{\mathbf{v}_j\}_{j=1}^m$ that spans the Krylov subspace, as well as two auxiliary scalar sequences $\{\alpha_j\}_{j=1}^m$ and $\{\beta_j\}_{j=2}^{m+1}$.

In the analysis, we will heavily rely on some useful properties of the Lanczos vectors $\{\mathbf{v}_j\}_{j=1}^m$, which we summarize in Proposition 3 for convenience.

Proposition 3. *The following statements hold true.*

(i) $\mathcal{K}_j(\mathbf{A}, \mathbf{b}) = \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_j\}$, *for any* $j \geq 1$.

(ii) $\mathbf{A}\mathbf{v}_j = \beta_{j+1}\mathbf{v}_{j+1} + \alpha_j \mathbf{v}_j + \beta_j \mathbf{v}_{j-1}$, *for any* $j \geq 1$.

(iii) Let $\mathbf{V}^{(j)} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_j] \in \mathbb{R}^{d \times j}$ *and* $\tilde{\mathbf{A}}^{(j)} = \text{tridiag}(\{\beta_l\}_{l=2}^j, \{\alpha_l\}_{l=1}^j, \{\beta_l\}_{l=2}^j)$ *defined as*

$$\begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \ddots & & \\ & & \ddots & \ddots & \beta_{j-1} & \\ & & & \beta_{j-1} & \alpha_{j-1} & \beta_j \\ & & & & \beta_j & \alpha_j \end{bmatrix}.$$

Then, we have that $(\mathbf{V}^{(j)})^\top \mathbf{A} \mathbf{V}^{(j)} = \tilde{\mathbf{A}}^{(j)}$ *and* $(\mathbf{V}^{(j)})^\top \mathbf{b} = \|\mathbf{b}\| \mathbf{e}_1^{(j)}$, *where* $\mathbf{e}_1^{(j)}$ *denotes the first standard basis vector in* $\mathbb{R}^j$.

Given the results of Proposition 3, by applying Algorithm 2 with $\mathbf{H}_k$ and $\mathbf{g}_k$ as the inputs, we can obtain

the orthogonal basis matrix $\mathbf{V}_k$ for the Krylov subspace $\mathcal{K}_m(\mathbf{H}_k, \mathbf{g}_k)$. As a byproduct, we also obtain the subspace gradient $\tilde{\mathbf{g}}_k = \mathbf{V}_k^\top \mathbf{g}_k$ and the subspace Hessian $\tilde{\mathbf{H}}_k = \mathbf{V}_k^\top \mathbf{H}_k \mathbf{V}_k$ by (iii) in Proposition 3. Moreover, utilizing the Krylov subspace has the additional benefit of having $\tilde{\mathbf{H}}_k$ as a tridiagonal matrix, which simplifies solving the cubic subproblem in (3) greatly. To elaborate, following the standard approach as described in Section 2.1, each Newton step requires solving a tridiagonal system of linear equations, which can be solved in $\mathcal{O}(m)$ operations. Thus, given $\tilde{\mathbf{g}}_k$ and $\tilde{\mathbf{H}}_k$, the cost of solving the cubic subproblem for our method is $\mathcal{O}(m)$.

Remark 1 (Computational cost of Algorithm 1). It is worth noting that in Algorithm 2, we only need to evaluate matrix-vector products of the matrix $\mathbf{A}$. Therefore, in the implementation of Algorithm 1, we do not have to store the Hessian matrix $\mathbf{H}_k$ explicitly, but only need to compute m Hessian-vector products. This computation can be done efficiently via back-propagation with a computational cost similar to gradient computation (Pearlmutter, 1994), which typically requires $\mathcal{O}(d)$ arithmetic operations. Hence, the total cost per iteration of our method is $\mathcal{O}(md)$.

4 CONVERGENCE ANALYSIS

In this section, we analyze the convergence rate of our proposed method in Algorithm 1. We first characterize the additional approximation error when solving the cubic subproblem over a Krylov subspace, as opposed to using the full-dimensional space $\mathbb{R}^d$ as in (2). This error analysis is key to our convergence proofs and is fully discussed in Section 4.1. We then present our main theorems in Section 4.2, covering both the cases where f is convex and strongly convex. Finally, in Section 4.3, we highlight the connection between our convergence bounds and the eigenspectrum of the Hessian matrices. In particular, we show that our convergence rate can be further improved when the Hessian matrices admit specific spectral structures.

4.1 Approximation Error of Krylov Subspace

As we discussed in Section 3.1, the crucial step in the convergence analysis is to establish a similar upper bound on $f(\mathbf{x}_{k+1})$ as in (4). We already showed that this key inequality will remain valid if Conditions (A), (B), and (C) are all satisfied, which is the case when the subspace $\mathcal{V}_k$ in Algorithm 1 is chosen as the maximal Krylov subspace. However, since we employ a Krylov subspace only up to dimension m in Algorithm 1, Condition (B) is potentially violated and this is the main source of the approximation error.

As it turns out, in our convergence analysis, it is enough to control the "minimal violation" of Condition (B) among all the Krylov subspaces up to the dimension m. To formalize this, let us define $\mathbf{V}_k^{(j)} \in \mathbb{R}^{d\times j}$ as the matrix that consists of the first j Lanczos vectors (cf. item (iii) of Proposition 2). Then, we can define $\mathbf{P}_k^{(j)} = \mathbf{V}_k^{(j)}\mathbf{V}_k^{(j)\top}$, which is the orthogonal projection matrix of the subspace $\mathcal{K}_j(\mathbf{H}_k, \mathbf{g}_k)$. In the following lemma, we provide the key inequality for Algorithm 1 that serves a similar role as the one in (4).

Lemma 1. *Let $\{\mathbf{x}_k\}_{k\geq 0}$ be the sequence generated by Algorithm 1 and suppose that Assumptions 1, 2 and 3 hold true. Suppose $M \geq L_2$. For any $\mathbf{s} \in \mathbb{R}^d$, we have*

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M}{6}\|\mathbf{s}\|^3 + \frac{1}{2}\min_{j\in\{1,\dots,m\}}\left\{(\mathbf{P}_k^{(j)}\mathbf{s})^\top \mathbf{H}_k \mathbf{P}_k^{(j)}\mathbf{s} - \mathbf{s}^\top \mathbf{H}_k \mathbf{s}\right\}.$$

In light of Lemma 1, our aim is to bound the additional error term $\min_{j\in\{1,\dots,m\}}\{(\mathbf{P}_k^{(j)}\mathbf{s})^\top \mathbf{H}_k \mathbf{P}_k^{(j)}\mathbf{s} - \mathbf{s}^\top \mathbf{H}_k \mathbf{s}\}$ for any $\mathbf{s} \in \mathbb{R}^d$. Before stating our result in Lemma 3, we introduce a quantity that plays a major role in our error analysis. Specifically, for a symmetric matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ and a vector $\mathbf{b} \in \mathbb{R}^d$, we define

$$\sigma^{(j)}(\mathbf{A}, \mathbf{b}) := \max_{\mathbf{w}\in\mathcal{K}^{(j)}, \|\mathbf{w}\|=1} \operatorname{dist}(\mathbf{A}\mathbf{w}, \mathcal{K}^{(j)})$$

for $j = 1, \dots, m$, where $\mathcal{K}^{(j)}$ denotes $\mathcal{K}_j(\mathbf{A}, \mathbf{b})$ and $\operatorname{dist}(\mathbf{u}, \mathcal{V}) := \min_{\mathbf{v}\in\mathcal{V}} \|\mathbf{u} - \mathbf{v}\|$ is the distance between a vector $\mathbf{u}$ and a subspace $\mathcal{V}$. Intuitively, $\sigma^{(j)}(\mathbf{A}, \mathbf{b})$ characterizes how far a vector $\mathbf{w}$ in $\mathcal{K}^{(j)}$ can be pushed away from the subspace under the linear mapping $\mathbf{A}$, and in particular we have $\sigma^{(j)}(\mathbf{A}, \mathbf{b}) = 0$ if and only if $\mathcal{K}_j(\mathbf{A}, \mathbf{b})$ is an invariant subspace with respect to the matrix $\mathbf{A}$. Furthermore, we define

$$\rho^{(m)}(\mathbf{A}, \mathbf{b}) := \left(\prod_{j=1}^{m} \sigma^{(j)}(\mathbf{A}, \mathbf{b})\right)^{\frac{1}{m}}. \tag{6}$$

As we shall discuss in Section 4.3, $\rho^{(m)}(\mathbf{A}, \mathbf{b})$ is closely related to the eigenspectrum of the matrix $\mathbf{A}$. In particular, it can be upper bounded by the largest eigenvalue of $\mathbf{A}$ in the worst case, as shown in Lemma 2.

Lemma 2. *Assume that $0 \preceq \mathbf{A} \preceq L_1\mathbf{I}$. Then, for any $\mathbf{b} \in \mathbb{R}^d$ we have $\rho^{(m)}(\mathbf{A}, \mathbf{b}) \leq 2^{1/m} L_1/4$.*

As a corollary, we emphasize that $\rho^{(m)}(\mathbf{A}, \mathbf{b})$ is independent of the problem's dimension and can be upper-bounded by the largest eigenvalue of $\mathbf{A}$. Moreover, this upper bound can be further improved if the matrix $\mathbf{A}$ exhibits specific spectral structures. We defer the discussions regarding these special cases to Section 4.3.

Next we leverage the definition in (6) to establish an upper bound on the approximation error caused by using a Krylov subspace of dimension m.

Lemma 3. *For any $k \geq 0$ and any $\mathbf{s} \in \mathbb{R}^d$, we have*

$$\min_{j\in\{1,\ldots,m\}} (\mathbf{P}_k^{(j)}\mathbf{s})^\top \mathbf{H}_k \mathbf{P}_k^{(j)}\mathbf{s} - \mathbf{s}^\top \mathbf{H}_k \mathbf{s} \leq \frac{2\rho^{(m)}(\mathbf{H}_k, \mathbf{g}_k)}{m}\|\mathbf{s}\|^2.$$

By combining Lemma 1 and Lemma 3, we obtain

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M}{6}\|\mathbf{s}\|^3 + \frac{\rho^{(m)}(\mathbf{H}_k, \mathbf{g}_k)}{m}\|\mathbf{s}\|^2. \quad (7)$$

Compared with (4), the inequality in (7) has an additional error term $\rho^{(m)}(\mathbf{H}_k, \mathbf{g}_k)\|\mathbf{s}\|^2/m$. Given that $\rho^{(m)}(\mathbf{H}_k, \mathbf{g}_k)$ can be upper bounded by a constant as shown in Lemma 2, we observe that the error term decays at the rate of $\mathcal{O}(1/m)$ as the subspace dimension m increases. Thus, as we should expect, a larger subspace dimension results in a smaller approximation error, which then leads to a faster convergence rate.

4.2 Main Results

In this section, we will leverage the upper bounds established in the previous section to present the convergence rate of our proposed method in both convex and strongly convex settings. To this end, let $\mathbf{x}^*$ be an optimal solution of f and define D as

$$D := \sup\{\|\mathbf{x} - \mathbf{x}^*\| : \mathbf{x} \in \mathbb{R}^d,\ f(\mathbf{x}) \leq f(\mathbf{x}_0)\}, \quad (8)$$

which is finite since the level-set $\{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ is bounded due to Assumption 2. In addition, the following quantity will be essential in measuring the convergence rate of our proposed algorithm

$$\rho_{\max}^{(m)} = \max_{i\in\{0,1,\ldots,k-1\}} \{\rho^{(m)}(\mathbf{H}_i, \mathbf{g}_i)\}, \quad (9)$$

as it provides a universal upper bound on the error corresponding to Condition (B). By Lemma 2, if we assume that $\nabla^2 f(\mathbf{x}_k) \preceq L_1 \mathbf{I}$ as in (Hanzely et al., 2020), then we have $\rho_{\max}^{(m)} \leq 2^{1/m} L_1/4$. On the other hand, $\rho_{\max}^{(m)}$ can be much smaller than L_1 in some special cases as discussed in Section 4.3. Next we present the convergence rate of our method in the convex setting.

Theorem 1. *Let $\{\mathbf{x}_k\}_{k\geq 0}$ be the sequence generated by Algorithm 1 and suppose that Assumptions 1, 2 and 3 hold true. Then, for any $k \geq 0$, we have*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{9\rho_{\max}^{(m)} D^2}{2m}\left(\frac{1}{k} + \frac{1}{k^2}\right) + \frac{9(L_2 + M)D^3}{2k^2}.$$

Theorem 1 shows that $f(\mathbf{x}_k) - f(\mathbf{x}^*)$ converges at a global convergence rate of $\mathcal{O}(\frac{1}{mk} + \frac{1}{k^2})$. As a corollary, to reach an error of $\epsilon > 0$, the number of required iterations can be upper bounded by $\mathcal{O}(\frac{1}{m\epsilon} + \frac{1}{\sqrt{\epsilon}})$, which is indeed independent of the problem's dimension d. To the best of our knowledge, this is the first subspace second-order method that attains a dimension-independent convergence rate. In comparison, SSCN in (Hanzely et al., 2020) achieves an iteration complexity of $\mathcal{O}(\frac{d-m}{m}\cdot\frac{1}{\epsilon} + \frac{d}{m}\cdot\frac{1}{\sqrt{\epsilon}})$. Note that when we deal with subspace second-order methods, we are particularly interested in scenarios where the subspace dimension m is a fixed constant much smaller than the problem dimension d, i.e., $m \ll d$. This is driven by a practical necessity to ensure a scalable per-iteration computational cost as the problem dimension increases. As our convergence bounds suggest, in the regime where $m \ll d$, our iteration complexity is lower than the one for SSCN by a factor of d.

Moreover, a better complexity can be achieved if f is strongly convex with parameter $\mu > 0$, that is, $\nabla^2 f(\mathbf{x}) \succeq \mu \mathbf{I}$ for all $\mathbf{x} \in \mathbb{R}^d$. As shown in Theorem 2, in this setting Algorithm 1 achieves a linear convergence rate.

Theorem 2. *Let $\{\mathbf{x}_k\}_{k\geq 0}$ be the sequence generated by Algorithm 1 and suppose Assumptions 1, 2, and 3 hold true. Moreover, assume that f is μ-strongly convex. Then, the number of iterations required to reach $\delta_k := f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$ can be upper bounded by*

$$k = \mathcal{O}\left(\left(\frac{\rho_{\max}^{(m)}}{m\mu} + 1\right)\log\frac{\delta_0}{\epsilon} + \frac{\sqrt{L_2 + M}\,\delta_0^{0.25}}{\mu^{0.75}}\right).$$

Remark 2. Similar to Hanzely et al. (2020), Theorem 2 continues to hold if we replace the strong convexity of f with the weaker assumption that $\frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^*\|^2 \leq f(\mathbf{x}) - f(\mathbf{x}^*)$ for any $\mathbf{x} \in \mathbb{R}^d$.

When the target accuracy ϵ is sufficiently small, Theorem 2 shows that the iteration complexity is dominated by $\frac{\rho_{\max}^{(m)}}{m\mu}\log\frac{1}{\epsilon}$, where the factor $\frac{\rho_{\max}^{(m)}}{m\mu}$ can be regarded as the effective condition number. In particular, consider the special case where $\mu\mathbf{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L_1\mathbf{I}$ for all $\mathbf{x} \in \mathbb{R}^d$. By Lemma 2, we always have $\frac{\rho_{\max}^{(m)}}{m\mu} \leq \frac{L_1}{m\mu}$. On the other hand, under the same setting as in (Hanzely et al., 2020), SSCN achieves an iteration complexity of $\mathcal{O}\big(\big(\frac{d-m}{m}\frac{L_1}{\mu} + \frac{d}{m}\big)\log\frac{1}{\epsilon}\big)$. Thus, we shave a factor of d from their iteration complexity bound when $m = \mathcal{O}(1)$. Moreover, as we discuss in the next section, the quantity $\rho_{\max}^{(m)}$ can be much smaller than L_1 when the Hessian admits some specific spectral structure, leading to an even faster convergence rate.

4.3 Special Cases

As shown in Theorems 1 and 2, the quantity $\rho_{\max}^{(m)}$ plays an important role in the convergence rate of our

proposed method. Moreover, by its definition in (9), we can upper bound $\rho^{(m)}(\mathbf{H}_i, \boldsymbol{g}_i)$ for $i = 0, 1, \ldots, k-1$ individually and then take the maximum to establish an upper bound on $\rho^{(m)}_{\max}$. In this section, we will show that $\rho^{(m)}(\mathbf{H}_i, \mathbf{g}_i)$ is closely related to the structure of the eigenspectrum of the Hessian matrix $\mathbf{H}_i$, and it can be much smaller than the worst-case upper bound in Lemma 2 in some special cases. The proofs in this section are deferred to Appendix D.

To begin with, we derive an alternative expression for $\rho^{(m)}(\mathbf{A}, \mathbf{b})$ in (6) in terms of matrix polynomials. Specifically, for a polynomial of degree m given by $p(x) = x^m + \sum_{i=0}^{m-1} c_i x^i$, we define $p(\mathbf{A}) := \mathbf{A}^m + \sum_{i=0}^{m-1} c_i \mathbf{A}^i$, where we observe the convention that $\mathbf{A}^0 = \mathbf{I}$. Moreover, we define $\mathcal{M}_m := \{x^m + \sum_{i=0}^{m-1} c_i x^i \mid c_0, \ldots, c_{m-1} \in \mathbb{R}\}$ as the set of monic polynomials of degree m. With these notations, we are ready to state Lemma 4.

Lemma 4. *The quantity $\rho^{(m)}(\mathbf{A}, \mathbf{b})$ in (6) can be equivalently expressed as*

$$\rho^{(m)}(\mathbf{A}, \mathbf{b}) = \min_{p \in \mathcal{M}_m} \left\| p(\mathbf{A}) \frac{\mathbf{b}}{\|\mathbf{b}\|} \right\|^{\frac{1}{m}}.$$

Lemma 4 implies $\rho^{(m)}(\mathbf{A}, \mathbf{b}) \leq \min_{p \in \mathcal{M}_m} \|p(\mathbf{A})\|_{\mathrm{op}}^{\frac{1}{m}}$, where $\|\cdot\|_{\mathrm{op}}$ denotes the operator norm of the matrix. Furthermore, assume that $\mathbf{A}$ has r distinct eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_r$, where $r \leq d$. Then, we have

$$\rho^{(m)}(\mathbf{A}, \mathbf{b}) \leq \min_{p \in \mathcal{M}_m} \max_{i \in \{1,2,\ldots,r\}} |p(\lambda_i)|^{\frac{1}{m}}.$$

Hence, by making suitable assumptions on the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_r$ and choosing the polynomial p accordingly, we can obtain different bounds on $\rho^{(m)}$.

Remark 3. It is worth noting that the matrix approximation problem $\min_{p \in \mathcal{M}_m} \|p(\mathbf{A})\|_{\mathrm{op}}$ has been studied before (Greenbaum and Trefethen, 1994), and the polynomial that achieves the minimum is also known as the *Chebyshev polynomial of* $\mathbf{A}$ (Toh and Trefethen, 1998; Faber et al., 2010). However, to the best of our knowledge, no explicit upper bound on its optimal value has been reported in the literature.

Example I. Our first result shows that $\rho^{(m)}$ can be upper bounded by the geometric mean of the largest m distinct eigenvalues of the Hessian $\mathbf{H}$.

Lemma 5. *Assume that the Hessian $\mathbf{H}$ has r distinct eigenvalues in decreasing order: $\lambda_1 > \lambda_2 > \cdots > \lambda_r$. For any $\mathbf{g} \in \mathbb{R}^d$, $\rho^{(m)}(\mathbf{H}, \mathbf{g}) \leq (\prod_{i=1}^m \lambda_i)^{\frac{1}{m}}$ when $m < r$ and $\rho^{(m)}(\mathbf{H}, \mathbf{g}) = 0$ when $m \geq r$.*

Hence, if the eigenvalues of $\mathbf{H}$ decay fast, then the bound in Lemma 5 can be much smaller than the one in Lemma 2. Moreover, as an important corollary, if the number of nonzero eigenvalues of the Hessian is at most $m-1$, then $\rho^{(m)}(\mathbf{H}, \mathbf{g}) = 0$ and the convergence rate of our algorithm becomes $\mathcal{O}(1/k^2)$. This means that, as a special case, if the rank of Hessian is at most $m-1$, the convergence rate of our Krylov CRN method with dimension m is as good as the full-dimensional CRN method. On the other hand, random subspace methods such as SSCN are unable to recover such a result when the Hessian has a low-rank structure.

Remark 4. The second result in Lemma 5 can be further strengthened: we have $\rho^{(m)}(\mathbf{H}, \mathbf{g}) = 0$ *if and only if* $m \geq r_0$, where r_0 denotes the dimension of the maximal Krylov subspace generated by $\mathbf{H}$ and $\mathbf{g}$. The proof can be found in Appendix D.3.

Example II. The second case we study is when the eigenvalues of the Hessian $\mathbf{H}$ are concentrated in two clusters $[0, \Delta]$ and $[L_1 - \Delta, L_1]$, where Δ is a small constant (see Goujaud et al. (2022)).

Lemma 6. *Assume that all the eigenvalues of the Hessian $\mathbf{H}$ lie within the two intervals $[0, \Delta]$ and $[L_1 - \Delta, L_1]$ for some $L_1 > \Delta > 0$. Then, when m is even, for any $\mathbf{g} \in \mathbb{R}^d$ we have $\rho^{(m)}(\mathbf{H}, \mathbf{g}) \leq 2^{1/m}\sqrt{\Delta(L_1 - \Delta)}/2$.*

Lemma 6 shows that if the eigenvalues of the Hessian are close to either 0 or L_1 with an error of size Δ, then the convergence rate of our method becomes $\mathcal{O}\left(\frac{\Delta(L_1-\Delta)}{mk} + \frac{1}{k^2}\right)$. As a result, to reach ϵ-accuracy, the overall iteration complexity of our method in the convex setting becomes $\mathcal{O}\left(\frac{\Delta(L_1-\Delta)}{\epsilon} + \frac{1}{\sqrt{\epsilon}}\right)$. Thus, if the size of the cluster satisfies $\Delta = \mathcal{O}(\sqrt{\epsilon})$, the overall iteration complexity of our method becomes $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\right)$, matching the one for the classic CRN method.

5 NUMERICAL EXPERIMENTS

In this section, we compare the numerical performance of our Krylov CRN method with full-dimensional CRN (Nesterov and Polyak, 2006) and SSCN (Hanzely et al., 2020). We focus on logistic regression problems on LIBSVM datasets (Chang and Lin, 2011). In our experiments, we use three different datasets representing low, medium, and high-dimensional problems. Specifically, we use "w8a", "rcv1", and "news20", with dimensions $d = 300$, $d = 47,236$, and $d = 1,355,191$, respectively. Note that the sample sizes n of these datasets are almost comparable with each other. We set the subspace dimension to be $m = 10$ in our method, while we vary the subspace dimension in SSCN from 10 to $1,000$ to ensure its best performance. Moreover, since the Lipschitz constant of the Hessian is unknown in advance, we use a backtracking line search scheme to select the regularization parameters. All ex-

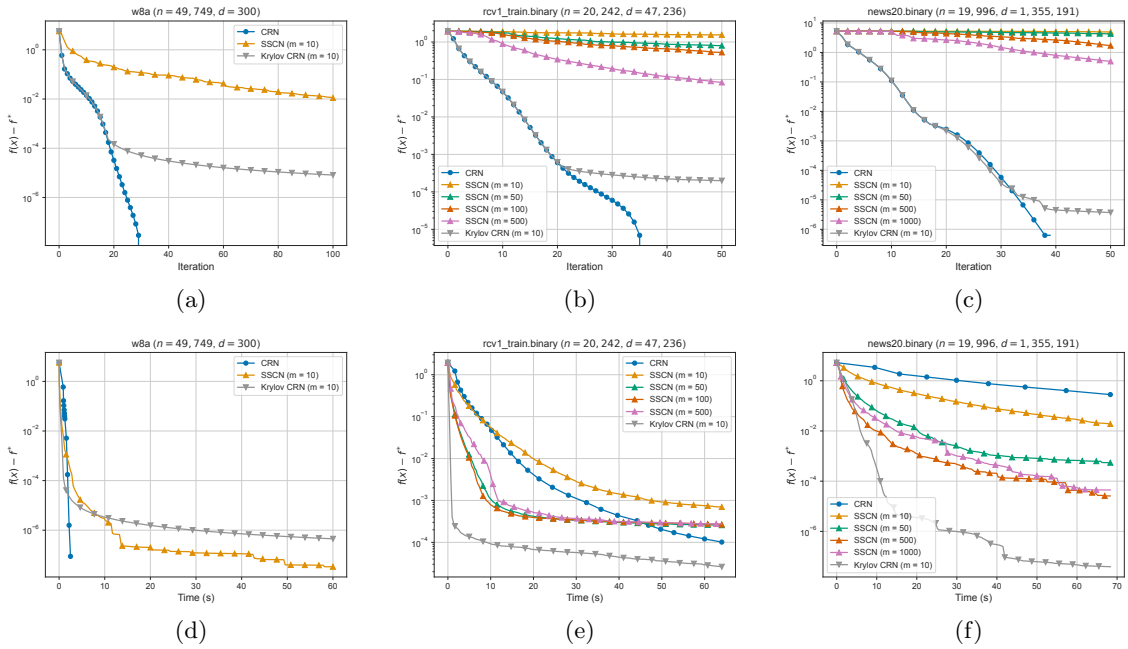


Figure 2: Numerical comparisons of CRN, SSCN, and our proposed Krylov CRN method for logistic regression on LIBSVM datasets. Here, m is the subspace dimension, n is the number of samples, and d is the number of parameters. The first row depicts the loss curves in terms of the number of iterations, while the second row depicts these curves in terms of the overall computation time. We observe that Krylov CRN converges much faster than the others when the dimension d is large.

periments are run on a MacBook Pro with an Apple M1 chip and 16GB RAM.

In Figures (a)-(c), we plot the suboptimality gap of CRN, SSCN, and our proposed Krylov CRN method in terms of the number of iterations. We observe that our method performs similarly to the full-dimensional CRN method, even though the cubic subproblem is solved over a subspace of dimension $m = 10$. Also, compared with SSCN, our Krylov CRN method is able to make much more progress in each iteration, and the gap between these two methods becomes more significant as the problem's dimension d increases. We should add that the convergence behaviors of these methods in Figures (a)-(c) in terms of iteration complexity are consistent with the theoretical results discussed in this paper. Specifically, we observe that SSCN converges slower as d increases, which is expected given the fact that its convergence rate scales linearly with d. Conversely, the convergence path of our proposed method remains almost unchanged as d increases. This consistency aligns with our theoretical findings that the iteration complexity of our method is dimension-free.

Since the per-iteration costs of these methods are different, in Figures (d)-(f) we also compare their performance in terms of the overall computation time. As shown in Figure (d), the full-dimensional CRN method converges very fast and outperforms all the other methods on a problem with a low-dimensional dataset, such as "w8a". However, as the dimension of the problem increases, the per-iteration cost of the CRN method rises rapidly and completes much fewer iterations in the given amount of time, resulting in its poor performance. Moreover, while SSCN typically incurs a smaller per-iteration cost than our method, Figures (e) and (f) show that our method still overall outperforms SSCN due to its faster convergence rate.

6 CONCLUSION

In this paper, we proposed the Krylov subspace cubic regularized Newton method, where we perform the cubic regularized Newton update over the Krylov subspace associated with the Hessian and the gradient at the current iterate. We proved that our proposed method converges at a rate of $\mathcal{O}\left(\frac{1}{mk} + \frac{1}{k^2}\right)$ in the convex setting, where m is the subspace dimension and k is the iteration counter. Furthermore, we characterized how the spectral structure of the Hessian matrices impacts its convergence, showing that the convergence rate can improve when the Hessian admits favorable structures. Finally, our experiments demonstrated the superior performance of our proposed method.

Acknowledgments

The work of R. Jiang was partially done while interning at Amazon Web Services. S. Sabach, M. Hong, and V. Cevher hold concurrent appointments as an Amazon Scholar and as a faculty at Technion, University of Minnesota, and EPFL, respectively. This paper describes their work performed at Amazon. The research of R. Jiang and A. Mokhtari is supported in part by NSF Grants 2007668, 2019844, and 2112471, ARO Grant W911NF2110226, the Machine Learning Lab (MLL) at UT Austin, and the Wireless Networking and Communications Group (WNCG) Industrial Affiliates Program.

References

Y. Carmon and J. C. Duchi. Analysis of Krylov subspace solutions of regularized non-convex quadratic problems. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Y. Carmon and J. C. Duchi. First-order methods for nonconvex quadratic minimization. *SIAM Review*, 62(2):395–436, 2020.

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

C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, 2011.

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

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

V. Faber, J. Liesen, and P. Tichý. On Chebyshev polynomials of matrices. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2205–2221, 2010.

B. Goujaud, D. Scieur, A. Dieuleveut, A. B. Taylor, and F. Pedregosa. Super-acceleration with cyclical step-sizes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3028–3065. PMLR, 2022.

R. Gower, D. Kovalev, F. Lieder, and P. Richtárik. RSN: randomized subspace Newton. In *Advances in Neural Information Processing Systems*, volume 32, pages 3028–3065. PMLR, 2019.

A. Greenbaum and L. N. Trefethen. GMRES/CR and Arnoldi/Lanczos as matrix approximation problems. *SIAM Journal on Scientific Computing*, 15(2): 359–368, 1994.

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

F. Hanzely, N. Doikov, Y. Nesterov, and P. Richtarik. Stochastic subspace cubic Newton method. In *International Conference on Machine Learning*, pages 4027–4038. PMLR, 2020.

S. Hanzely. Sketch-and-project meets Newton method: Global $\mathcal{O}(k^{-2})$ convergence with low-rank updates. *arXiv preprint arXiv:2305.13082*, 2023.

C. Lanczos. An iteration method for the solution of the eigenvalues problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950.

Z.-Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.

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

Y. Nesterov. *Lectures on Convex Optimization*. Springer International Publishing, 2018.

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

B. N. Parlett. *The symmetric eigenvalue problem*. SIAM, 1998.

B. A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural computation*, 6(1):147–160, 1994.

Y. Saad. *Numerical methods for large eigenvalue problems: revised edition*. SIAM, 2011.

K.-C. Toh and L. N. Trefethen. The chebyshev polynomials of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 20(2):400–419, 1998.

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

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes. We specify the assumptions in Section 2 and our algorithms are described in Algorithms 1 and 2.]

(b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes. The proofs are provided in the supplementary material.]

(c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes. The source code is included in the supplementary material.]

2. For any theoretical claim, check if you include:

(a) Statements of the full set of assumptions of all theoretical results. [Yes. In Theorems 1 and 2, we clearly state the assumptions.]

(b) Complete proofs of all theoretical results. [Yes. The proofs are included in the supplementary material.]

(c) Clear explanations of any assumptions. [Yes. As we mentioned in Section 2, the assumptions are common in the analysis of CRN-type methods.]

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes. The code is included in the supplementary material.]

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes. The implementation details can be found in the supplementary material.]

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable. The proposed method is a deterministic algorithm.]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Yes]

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

A PROOF OF PROPOSITION 2

To prove item (i), we note that Condition (A) is equivalent to $(\mathbf{P}_k\mathbf{g}_k - \mathbf{g}_k)^\top \mathbf{s} \le 0$ for any $\mathbf{s} \in \mathbb{R}^d$. In particular, by choosing $\mathbf{s} = \mathbf{P}_k\mathbf{g}_k - \mathbf{g}_k$, we can deduce that $\mathbf{g}_k = \mathbf{P}_k\mathbf{g}_k$. Since $\mathbf{P}_k$ is an orthogonal projection matrix associated with the subspace $\mathcal{V}_k$, this holds if and only if $\mathbf{g}_k \in \mathcal{V}_k$.

To prove item (ii), we consider the following decomposition $\mathbf{s} = \hat{\mathbf{s}} + \mathbf{s}^\perp$, where $\hat{\mathbf{s}} \in \mathcal{V}_k$ and $\mathbf{s}^\perp \in \mathcal{V}_k^\perp$. Here, $\mathcal{V}_k^\perp$ denotes the orthogonal complement of $\mathcal{V}_k$. Since $\hat{\mathbf{s}} = \mathbf{P}_k\mathbf{s}$, we have

$$\begin{aligned}\mathbf{s}^\top\mathbf{H}_k\mathbf{s} - (\mathbf{P}_k\mathbf{s})^\top\mathbf{H}_k(\mathbf{P}_k\mathbf{s}) = \mathbf{s}^\top\mathbf{H}_k\mathbf{s} - \hat{\mathbf{s}}^\top\mathbf{H}_k\hat{\mathbf{s}} &= (\hat{\mathbf{s}} + \mathbf{s}^\perp)^\top\mathbf{H}_k(\hat{\mathbf{s}} + \mathbf{s}^\perp) - \hat{\mathbf{s}}^\top\mathbf{H}_k\hat{\mathbf{s}} \\ &= 2(\mathbf{s}^\perp)^\top\mathbf{H}_k\hat{\mathbf{s}} + (\mathbf{s}^\perp)^\top\mathbf{H}_k\mathbf{s}^\perp.\end{aligned}$$

Thus, Condition (B) is equivalent to

$$2(\mathbf{s}^\perp)^\top\mathbf{H}_k\hat{\mathbf{s}} + (\mathbf{s}^\perp)^\top\mathbf{H}_k\mathbf{s}^\perp \ge 0, \quad \forall \hat{\mathbf{s}} \in \mathcal{V}_k \text{ and } \mathbf{s}^\perp \in \mathcal{V}_k^\perp. \tag{10}$$

Moreover, we claim that (10) further implies

$$(\mathbf{s}^\perp)^\top\mathbf{H}_k\hat{\mathbf{s}} = 0, \quad \forall \hat{\mathbf{s}} \in \mathcal{V}_k \text{ and } \mathbf{s}^\perp \in \mathcal{V}_k^\perp. \tag{11}$$

Otherwise, suppose there exist $\hat{\mathbf{s}}_0 \in \mathcal{V}_k$ and $\mathbf{s}_0^\perp \in \mathcal{V}_k^\perp$ with $(\mathbf{s}_0^\perp)^\top\mathbf{H}_k\hat{\mathbf{s}}_0 \neq 0$. Without loss of generality, assume $(\mathbf{s}_0^\perp)^\top\mathbf{H}_k\hat{\mathbf{s}}_0 > 0$. Since $\mathcal{V}_k$ is a linear subspace, we have $\lambda\hat{\mathbf{s}}_0 \in \mathcal{V}_k$ for any $\lambda \in \mathbb{R}$. However, when $\lambda \to -\infty$, we have

$$2(\mathbf{s}_0^\perp)^\top\mathbf{H}_k(\lambda\hat{\mathbf{s}}_0) + (\mathbf{s}_0^\perp)^\top\mathbf{H}_k\mathbf{s}_0^\perp = 2\lambda(\mathbf{s}_0^\perp)^\top\mathbf{H}_k\hat{\mathbf{s}}_0 + (\mathbf{s}_0^\perp)^\top\mathbf{H}_k\mathbf{s}_0^\perp \to -\infty,$$

which leads to a contradiction with (10). This proves that the statement in (11). Finally, since $\mathbf{s}^\perp$ can be any vector in $\mathcal{V}_k^\perp$, (11) holds if and only if $\mathbf{H}_k\hat{\mathbf{s}} \in \mathcal{V}_k$. Since $\hat{\mathbf{s}}$ is an arbitrary vector in $\mathcal{V}_k$, by definition, this shows that $\mathcal{V}_k$ is an invariant subspace.

B THE LANCZOS METHOD

First, we provide the proof of Proposition 3 in Section B.1. Then in Section B.2, we present additional technical lemmas that we will use in the convergence proof.

B.1 Proof of Proposition 3

Item (i) is a standard property of the Lanczos method; for instance, see Saad (2011, Proposition 6.5). Item (ii) follows from Lines 4, 6, and 8 in the update rule of Algorithm 2. To prove item (iii), we first rewrite the recursive formula in item (ii) in a matrix form, as shown in the following lemma.

Lemma 7. *We have* $\mathbf{A}\mathbf{V}^{(j)} = \mathbf{V}^{(j)}\tilde{\mathbf{A}}^{(j)} + \beta_{j+1}\mathbf{v}_{j+1}(\mathbf{e}_j^{(j)})^\top$, *where* $\mathbf{e}_j^{(j)}$ *is the* j*-th standard basis vector in* $\mathbb{R}^j$.

Proof. It suffices to verify that the two matrices in the identity have the exactly same columns. For any $l \in \{1, \ldots, j\}$, the l-th column of $\mathbf{A}\mathbf{V}^{(j)}$ is given by $\mathbf{A}\mathbf{v}_l$. On the other hand, by the definition of $\tilde{\mathbf{A}}^{(j)}$, we can observe that the l-th column of the the right-hand side matrix is given by $\beta_l\mathbf{v}_{l-1} + \alpha_l\mathbf{v}_l + \beta_{l+1}\mathbf{v}_{l+1}$. Hence, the equality holds due to Item (ii) in Proposition 3. □

Now we are ready to prove item (iii) in Proposition 3. Since the vectors $\{\mathbf{v}_j\}_{j=1}^m$ are orthonormal, it holds that $(\mathbf{V}^{(j)})^\top\mathbf{V}^{(j)} = \mathbf{I}$ and $\left(\mathbf{V}^{(j)}\right)^\top\mathbf{v}_{j+1} = 0$. Thus, by Lemma 7, we have $(\mathbf{V}^{(j)})^\top\mathbf{A}\mathbf{V}^{(j)} = (\mathbf{V}^{(j)})^\top\mathbf{V}^{(j)}\tilde{\mathbf{A}}^{(j)} + \beta_{j+1}(\mathbf{V}^{(j)})^\top\mathbf{v}_{j+1}(\mathbf{e}_j^{(j)})^\top = \tilde{\mathbf{A}}^{(j)}$. Moreover, note that we set $\mathbf{v}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|}$ in Algorithm 2. Since $(\mathbf{V}^{(j)})^\top\mathbf{v}_1 = \mathbf{e}_1^{(j)}$, we further obtain that $(\mathbf{V}^{(j)})^\top\mathbf{b} = \|\mathbf{b}\|\mathbf{e}_1^{(j)}$. This completes the proof.

B.2 Technical Lemmas

In this section, we provide some additional useful results regarding the outputs of the Lanczos method in Algorithm 2. For the convenience of the reader, we briefly recap our notations. Given the input matrix $\mathbf{A}$ and the input vector $\mathbf{b}$, we let $\{\mathbf{v}_j\}_{j=1}^m$, $\{\alpha_j\}_{j=1}^m$, $\{\beta_j\}_{j=2}^{m+1}$ be the sequences generated during the execution of Algorithm 2. Also recall that $\mathbf{V}^{(j)} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_j] \in \mathbb{R}^{d\times j}$ and $\mathbf{P}^{(j)} = \mathbf{V}^{(j)}(\mathbf{V}^{(j)})^\top$.

Lemma 8. *The following statement holds true.*

(i) $(\mathbf{I} - \mathbf{P}^{(j)})\mathbf{A}\mathbf{P}^{(j)} = \beta_{j+1}\mathbf{v}_{j+1}\mathbf{v}_j^\top$ *for any* $j \geq 1$.

(ii) For any $j \geq 1$, *we have*

$$\beta_{j+1} = \sigma^{(j)}(\mathbf{A}, \mathbf{b}) := \max_{\mathbf{w}\in\mathcal{K}^{(j)}, \|\mathbf{w}\|=1} \operatorname{dist}(\mathbf{A}\mathbf{w}, \mathcal{K}^{(j)}), \tag{12}$$

where $\mathcal{K}^{(j)}$ *denotes* $\mathcal{K}_j(\mathbf{A}, \mathbf{b})$ *and* $\operatorname{dist}(\mathbf{u}, \mathcal{V}) := \min_{\mathbf{v}\in\mathcal{V}} \|\mathbf{u} - \mathbf{v}\|$ *is the distance between a vector* $\mathbf{u}$ *and a subspace* $\mathcal{V}$.

(iii) We have

$$\prod_{j=2}^{m+1} \beta_j = \min_{\mathbf{u}\in\mathcal{K}_m(\mathbf{A},\mathbf{b})} \|\mathbf{A}^m\mathbf{v}_1 - \mathbf{u}\|. \tag{13}$$

We note that some results in Lemma 8 are likely known in the literature (see, e.g., Parlett (1998)), but for completeness we provide their proofs below.

Proof. To prove item (i), by using Lemma 7 and the fact that $\mathbf{P}^{(j)} = \mathbf{V}^{(j)}(\mathbf{V}^{(j)})^\top$, we obtain

$$\mathbf{A}\mathbf{P}^{(j)} = \mathbf{A}\mathbf{V}^{(j)}(\mathbf{V}^{(j)})^\top = \mathbf{V}^{(j)}\tilde{\mathbf{A}}^{(j)}(\mathbf{V}^{(j)})^\top + \beta_{j+1}\mathbf{v}_{j+1}(\mathbf{e}_j^{(j)})^\top(\mathbf{V}^{(j)})^\top = \mathbf{V}^{(j)}\tilde{\mathbf{A}}^{(j)}(\mathbf{V}^{(j)})^\top + \beta_{j+1}\mathbf{v}_{j+1}\mathbf{v}_j^\top.$$

Moreover, since the vectors $\{\mathbf{v}_j\}_{j=1}^m$ are orthonormal, it holds that $(\mathbf{I}-\mathbf{P}^{(j)})\mathbf{V}^{(j)} = \mathbf{V}^{(j)} - \mathbf{V}^{(j)}(\mathbf{V}^{(j)})^\top\mathbf{V}^{(j)} = 0$ and $(\mathbf{I}-\mathbf{P}^{(j)})\mathbf{v}_{j+1} = \mathbf{v}_{j+1} - (\mathbf{V}^{(j)})^\top\mathbf{V}^{(j)}\mathbf{v}_{j+1} = \mathbf{v}_{j+1}$. Hence, we conclude that $(\mathbf{I}-\mathbf{P}^{(j)})\mathbf{A}\mathbf{P}^{(j)} = \beta_{j+1}\mathbf{v}_{j+1}\mathbf{v}_j^\top$.

To prove item (ii), we will use the result we just proved in item (i). Since $\mathbf{P}^{(j)}$ is the orthogonal projection matrix associated to $\mathcal{K}_j(\mathbf{A}, \mathbf{b})$, we have $\operatorname{dist}(\mathbf{A}\mathbf{w}, \mathcal{K}^{(j)}) = \|(\mathbf{I} - \mathbf{P}^{(j)})\mathbf{A}\mathbf{w}\|$ by the distance-minimizing property of orthogonal projection. Moreover, note that for any $\mathbf{w} \in \mathcal{K}_j(\mathbf{A}, \mathbf{b})$, we have $\mathbf{P}^{(j)}\mathbf{w} = \mathbf{w}$. Thus, we can write

$$\operatorname{dist}(\mathbf{A}\mathbf{w}, \mathcal{K}^{(j)}) = \|(\mathbf{I} - \mathbf{P}^{(j)})\mathbf{A}\mathbf{w}\| = \|(\mathbf{I} - \mathbf{P}^{(j)})\mathbf{A}\mathbf{P}^{(j)}\mathbf{w}\| = \beta_{j+1}\|\mathbf{v}_{j+1}(\mathbf{v}_j^\top\mathbf{w})\| = \beta_{j+1}|\mathbf{v}_j^\top\mathbf{w}|.$$

Thus, the maximum in (12) is achieved when $\mathbf{w} = \pm\mathbf{v}_j$, which proves the desired result.

To prove item (iii), we will first use (12) to derive the following alternative expression for β_{j+1}:

$$\beta_2 = \min_{\mathbf{u}\in\mathcal{K}_1(\mathbf{A},\mathbf{b})} \|\mathbf{A}\mathbf{v}_1 - \mathbf{u}\| \quad \text{and} \quad \beta_{j+1} = \frac{\min_{\mathbf{u}\in\mathcal{K}_j(\mathbf{A},\mathbf{b})} \|\mathbf{A}^j\mathbf{v}_1 - \mathbf{u}\|}{\min_{\mathbf{u}\in\mathcal{K}_{j-1}(\mathbf{A},\mathbf{b})} \|\mathbf{A}^{j-1}\mathbf{v}_1 - \mathbf{u}\|} \text{ for } j \geq 2.$$

If this is the case, then (13) directly follows from telescoping. To prove this, we first note that $\mathcal{K}_1(\mathbf{A}, \mathbf{b}) = \operatorname{span}(\mathbf{v}_1)$ and thus

$$\beta_2 = \max_{\mathbf{w}\in\mathcal{K}_1(\mathbf{A},\mathbf{b}), \|\mathbf{w}\|=1} \operatorname{dist}(\mathbf{A}\mathbf{w}, \mathcal{K}_1(\mathbf{A}, \mathbf{b})) = \operatorname{dist}(\mathbf{A}\mathbf{v}_1, \mathcal{K}_1(\mathbf{A}, \mathbf{b})) = \min_{\mathbf{u}\in\mathcal{K}_1(\mathbf{A},\mathbf{b})} \|\mathbf{A}\mathbf{v}_1 - \mathbf{u}\|.$$

Moreover, since $\mathcal{K}_j(\mathbf{A}, \mathbf{b})$ is a linear subspace, we have $\operatorname{dist}(\lambda\mathbf{A}\mathbf{w}, \mathcal{K}_j(\mathbf{A}, \mathbf{b})) = \lambda\operatorname{dist}(\mathbf{A}\mathbf{w}, \mathcal{K}_j(\mathbf{A}, \mathbf{b}))$ for any $\lambda > 0$. Thus, we can equivalently write (12) as

$$\beta_{j+1} = \max_{\mathbf{w}\in\mathcal{K}_j(\mathbf{A},\mathbf{b})} \frac{\operatorname{dist}(\mathbf{A}\mathbf{w}, \mathcal{K}_j(\mathbf{A}, \mathbf{b}))}{\|\mathbf{w}\|}. \tag{14}$$

Note that for any $\mathbf{w} \in \mathcal{K}_j(\mathbf{A}, \mathbf{b})$, it can be decomposed as $\mathbf{w} = c_0\mathbf{A}^{j-1}\mathbf{v}_1 - \mathbf{u}$, where $c_0 \in \mathbb{R}$ is some scalar and $\mathbf{u} \in \mathcal{K}_{j-1}(\mathbf{A}, \mathbf{b})$. If $c_0 = 0$, then $\mathbf{w} \in \mathcal{K}_{j-1}(\mathbf{A}, \mathbf{b})$ and further we have $\mathbf{A}\mathbf{w} \in \mathcal{K}_j(\mathbf{A}, \mathbf{b})$, which implies that

$\operatorname{dist}(\mathbf{A}\mathbf{w}, \mathcal{K}^{(j)}) = 0$. Hence, for the maximization problem in (14), without loss of generality, we can assume that $\mathbf{w} = \mathbf{A}^{j-1}\mathbf{v}_1 - \mathbf{u}$ with $\mathbf{u} \in \mathcal{K}_{j-1}(\mathbf{A}, \mathbf{b})$ and rewrite (14) as

$$\beta_{j+1} = \max_{\mathbf{u}\in\mathcal{K}_{j-1}(\mathbf{A},\mathbf{b})} \frac{\operatorname{dist}(\mathbf{A}^j\mathbf{v}_1 - \mathbf{A}\mathbf{u}, \mathcal{K}_j(\mathbf{A},\mathbf{b}))}{\|\mathbf{A}^{j-1}\mathbf{v}_1 - \mathbf{u}\|} = \max_{\mathbf{u}\in\mathcal{K}_{j-1}(\mathbf{A},\mathbf{b})} \frac{\operatorname{dist}(\mathbf{A}^j\mathbf{v}_1, \mathcal{K}_j(\mathbf{A},\mathbf{b}))}{\|\mathbf{A}^{j-1}\mathbf{v}_1 - \mathbf{u}\|}, \tag{15}$$

where we used the fact that $\mathbf{A}\mathbf{u} \in \mathcal{K}_j(\mathbf{A}, \mathbf{b})$ in the second equality. Moreover, the nominator in (15) can be written as $\operatorname{dist}(\mathbf{A}^j\mathbf{v}_1, \mathcal{K}_j(\mathbf{A},\mathbf{b})) = \min_{\mathbf{u}\in\mathcal{K}_j(\mathbf{A},\mathbf{b})} \|\mathbf{A}^j\mathbf{v}_1 - \mathbf{u}\|$, while the denominator is equal to $\min_{\mathbf{u}\in\mathcal{K}_{j-1}(\mathbf{A},\mathbf{b})} \|\mathbf{A}^{j-1}\mathbf{v}_1 - \mathbf{u}\|$. This completes the proof. □

As a corollary of item (ii) in Lemma 8, the quantity $\rho^{(m)}(\mathbf{A}, \mathbf{b})$ defined in (6) can also be written as $\rho^{(m)}(\mathbf{A}, \mathbf{b}) = \left(\prod_{j=1}^m \beta_{j+1}\right)^{\frac{1}{m}}$. This observation will be used in the following lemma.

Lemma 9. *Assume that $\mathbf{A} \succeq 0$. Then, for any $\mathbf{s} \in \mathbb{R}^d$, we have*

$$\min_{j\in\{1,2,\ldots,m\}} \left\{(\mathbf{P}^{(j)}\mathbf{s})^\top \mathbf{A}\mathbf{P}^{(j)}\mathbf{s} - \mathbf{s}^\top\mathbf{A}\mathbf{s}\right\} \le \frac{2\rho^{(m)}(\mathbf{A},\mathbf{b})}{m}\|\mathbf{s}\|^2. \tag{16}$$

Proof. To begin with, fix $j \in \{1, 2, \ldots, m\}$. By using $\mathbf{s} = \mathbf{P}^{(j)}\mathbf{s} + (\mathbf{I} - \mathbf{P}^{(j)})\mathbf{s}$, we can write

$$\mathbf{s}^\top\mathbf{A}\mathbf{s} = (\mathbf{P}^{(j)}\mathbf{s})^\top\mathbf{A}\mathbf{P}^{(j)}\mathbf{s} + 2\left((\mathbf{I} - \mathbf{P}^{(j)})\mathbf{s}\right)^\top \mathbf{A}\mathbf{P}^{(j)}\mathbf{s} + \left((\mathbf{I} - \mathbf{P}^{(j)})\mathbf{s}\right)^\top \mathbf{A}(\mathbf{I} - \mathbf{P}^{(j)})\mathbf{s}.$$

Since $\mathbf{A} \succeq 0$, we have $\left((\mathbf{I} - \mathbf{P}^{(j)})\mathbf{s}\right)^\top \mathbf{A}(\mathbf{I} - \mathbf{P}^{(j)})\mathbf{s} \ge 0$, which implies that

$$(\mathbf{P}^{(j)}\mathbf{s})^\top\mathbf{A}\mathbf{P}^{(j)}\mathbf{s} - \mathbf{s}^\top\mathbf{A}\mathbf{s} \le -2\left((\mathbf{I} - \mathbf{P}^{(j)})\mathbf{s}\right)^\top \mathbf{A}\mathbf{P}^{(j)}\mathbf{s}. \tag{17}$$

To bound the right-hand side of (17), we first note that $\left((\mathbf{I} - \mathbf{P}^{(j)})\mathbf{s}\right)^\top \mathbf{A}\mathbf{P}^{(j)}\mathbf{s} = \mathbf{s}^\top(\mathbf{I} - \mathbf{P}^{(j)})\mathbf{A}\mathbf{P}^{(j)}\mathbf{s} = \beta_{j+1}(\mathbf{v}_{j+1}^\top\mathbf{s})(\mathbf{v}_j^\top\mathbf{s})$, where we used item (i) in Lemma 8 to obtain the last equality. Hence, (17) becomes

$$(\mathbf{P}^{(j)}\mathbf{s})^\top\mathbf{A}\mathbf{P}^{(j)}\mathbf{s} - \mathbf{s}^\top\mathbf{A}\mathbf{s} \le -2\beta_{j+1}(\mathbf{v}_{j+1}^\top\mathbf{s})(\mathbf{v}_j^\top\mathbf{s}) \le 2\beta_{j+1}|\mathbf{v}_j^\top\mathbf{s}||\mathbf{v}_{j+1}^\top\mathbf{s}|.$$

Since the inequality above holds for any j, by taking the minimum over $j = 1, 2, \ldots, m$, it further leads to

$$\min_{j\in\{1,2,\ldots,m\}} \left\{(\mathbf{P}^{(j)}\mathbf{s})^\top\mathbf{A}\mathbf{P}^{(j)}\mathbf{s} - \mathbf{s}^\top\mathbf{A}\mathbf{s}\right\} \le 2\min_{j\in\{1,2,\ldots,m\}} \beta_{j+1}|\mathbf{v}_j^\top\mathbf{s}||\mathbf{v}_{j+1}^\top\mathbf{s}|. \tag{18}$$

Since the minimum is upper bounded by the geometric mean, we also have

$$\min_{j\in\{1,2,\ldots,m\}} \beta_{j+1}|\mathbf{v}_j^\top\mathbf{s}||\mathbf{v}_{j+1}^\top\mathbf{s}| \le \left(\prod_{j=1}^m \beta_{j+1}|\mathbf{v}_j^\top\mathbf{s}||\mathbf{v}_{j+1}^\top\mathbf{s}|\right)^{\frac{1}{m}} \le \left(\prod_{j=1}^m \beta_{j+1}\right)^{\frac{1}{m}} \left(\prod_{j=1}^m |\mathbf{v}_j^\top\mathbf{s}||\mathbf{v}_{j+1}^\top\mathbf{s}|\right)^{\frac{1}{m}}. \tag{19}$$

Recall that we have $\rho^{(m)}(\mathbf{A}, \mathbf{b}) = \left(\prod_{j=1}^m \beta_{j+1}\right)^{\frac{1}{m}}$. Furthermore, note that $\{\mathbf{v}_j\}_{j=1}^m$ are orthonormal and by Bessel's inequality we have

$$\sum_{j=1}^{m+1} |\mathbf{v}_j^\top\mathbf{s}|^2 \le \|\mathbf{s}\|^2.$$

Using the inequality of arithmetic and geometric means, we obtain

$$\left(\prod_{j=1}^m |\mathbf{v}_j^\top\mathbf{s}||\mathbf{v}_{j+1}^\top\mathbf{s}|\right)^{\frac{1}{m}} \le \frac{1}{m}\sum_{i=1}^m |\mathbf{v}_j^\top\mathbf{s}||\mathbf{v}_{j+1}^\top\mathbf{s}| \le \frac{1}{m}\sum_{i=1}^m \frac{1}{2}(|\mathbf{v}_j^\top\mathbf{s}|^2 + |\mathbf{v}_{j+1}^\top\mathbf{s}|^2) \le \frac{1}{m}\sum_{j=1}^{m+1} |\mathbf{v}_j^\top\mathbf{s}|^2 \le \frac{1}{m}\|\mathbf{s}\|^2. \tag{20}$$

Finally (16) follows from (18), (19) and (20). This completes the proof. □

C PROOFS OF THE MAIN THEOREMS

To begin with, recall that the columns of $\mathbf{V}_k$ form an orthogonal basis for the Krylov subspace $\mathcal{K}_m(\mathbf{H}_k, \mathbf{g}_k)$. Hence, for any $\mathbf{s} \in \mathcal{K}_m(\mathbf{H}_k, \mathbf{g}_k)$, there exists $\mathbf{z} \in \mathbb{R}^m$ such that $\mathbf{s} = \mathbf{V}_k\mathbf{z}$ and $\|\mathbf{z}\| = \|\mathbf{s}\|$. Thus, by substituting $\mathbf{V}_k\mathbf{z}$ for $\mathbf{s}$ in (3), the update rule of Algorithm 1 can be equivalently written as

$$\mathbf{s}_k := \mathop{\arg\min}_{\mathbf{s}\in\mathcal{K}_m(\mathbf{H}_k,\mathbf{g}_k)} \left\{ \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M}{6}\|\mathbf{s}\|^3 \right\} \tag{21}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k. \tag{22}$$

This alternative form of Algorithm 1 will be useful in the subsequent proofs.

C.1 Proof of Lemma 1

As the first step of proving Lemma 1, we first present the following lemma, which shows a similar inequality as in (4) but only holds for vector $\mathbf{s}$ in the subspace $\mathcal{K}_m(\mathbf{H}_k, \mathbf{g}_k)$.

Lemma 10. *For any $\mathbf{s} \in \mathcal{K}_m(\mathbf{H}_k, \mathbf{g}_k)$, we have*

$$f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M}{6}\|\mathbf{s}\|^3.$$

Proof. Since $M \ge L_2$ and $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, by applying Proposition 1 with $\mathbf{x}' = \mathbf{x}_k$ and $\mathbf{x} = \mathbf{x}_{k+1}$ we have

$$f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^\top \mathbf{H}_k \mathbf{s}_k + \frac{M}{6}\|\mathbf{s}_k\|^3.$$

According to (21), $\mathbf{s}_k$ is chosen as the minimizer of the cubic subproblem over the subspace $\mathcal{K}_m(\mathbf{H}_k, \mathbf{g}_k)$, which means that $\mathbf{g}_k^\top \mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^\top \mathbf{H}_k \mathbf{s}_k + \frac{M}{6}\|\mathbf{s}_k\|^3 \le \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M}{6}\|\mathbf{s}\|^3$ for any $\mathbf{s} \in \mathcal{K}_m(\mathbf{H}_k, \mathbf{g}_k)$. Hence, the result immediately follows. □

Now we are ready to prove Lemma 1.

Proof of Lemma 1. Recall that $\mathbf{P}_k^{(j)} = \mathbf{V}_k^{(j)}\mathbf{V}_k^{(j)\top}$ is the orthogonal projection matrix of the subspace $\mathcal{K}_j(\mathbf{H}_k, \mathbf{g}_k)$. For any given $\mathbf{s} \in \mathbb{R}^d$, define $\mathbf{s}^{(j)} = \mathbf{P}_k^{(j)}\mathbf{s}$ for $j = 1, 2, \ldots, m$. Since we have $\mathbf{s}^{(j)} \in \mathcal{K}_j(\mathbf{H}_k, \mathbf{g}_k) \subset \mathcal{K}_m(\mathbf{H}_k, \mathbf{g}_k)$ for any $j \le m$, by applying Lemma 10 with $\mathbf{s} = \mathbf{s}^{(j)}$, we get

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\le f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s}^{(j)} + \frac{1}{2}(\mathbf{s}^{(j)})^\top \mathbf{H}_k \mathbf{s}^{(j)} + \frac{M}{6}\|\mathbf{s}^{(j)}\|^3 \\ &= f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{P}_k^{(j)}\mathbf{s} + \frac{1}{2}(\mathbf{P}_k^{(j)}\mathbf{s})^\top \mathbf{H}_k \mathbf{P}_k^{(j)}\mathbf{s} + \frac{M}{6}\|\mathbf{P}_k^{(j)}\mathbf{s}\|^3. \end{aligned}$$

Moreover, since $\mathbf{g}_k \in \mathcal{K}_j(\mathbf{H}_k, \mathbf{g}_k)$ for all $j \ge 0$, we have $\mathbf{P}_k^{(j)}\mathbf{g}_k = \mathbf{g}_k$. Also, since $\mathbf{P}_k^{(j)}$ is an orthogonal projection matrix, we have $\|\mathbf{P}_k^{(j)}\mathbf{s}\| \le \|\mathbf{s}\|$ for any $\mathbf{s} \in \mathbb{R}^d$. Thus, we get

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\le f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}(\mathbf{P}_k^{(j)}\mathbf{s})^\top \mathbf{H}_k \mathbf{P}_k^{(j)}\mathbf{s} + \frac{M}{6}\|\mathbf{s}\|^3 \\ &= f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M}{6}\|\mathbf{s}\|^3 + \frac{1}{2}\left((\mathbf{P}_k^{(j)}\mathbf{s})^\top \mathbf{H}_k \mathbf{P}_k^{(j)}\mathbf{s} - \mathbf{s}^\top \mathbf{H}_k \mathbf{s} \right). \end{aligned} \tag{23}$$

Since (23) holds for any $j \in \{1, 2, \ldots, m\}$, we obtain the desired result by taking the minimum of the right-hand side over $j = 1, \ldots, m$. □

C.2 Proof of Lemma 3

Since f is convex by Assumption 1, we have $\mathbf{H}_k \succeq 0$. Thus, Lemma 3 immediately follows from Lemma 9 by setting $\mathbf{A} = \mathbf{H}_k$ and $\mathbf{b} = \mathbf{g}_k$.

C.3 Proof of Theorem 1

Our starting point is the inequality in (7). By choosing $\mathbf{s} = 0$, we immediately obtain that $f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k)$, which implies $f(\mathbf{x}_k) \le f(\mathbf{x}_0)$ for all $k \ge 0$. Thus, $\mathbf{x}_k$ is in the level-set $\{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) \le f(\mathbf{x}_0)\}$ and hence $\|\mathbf{x}_k - \mathbf{x}^*\| \le D$ according to (8). Moreover, by Proposition 1, it holds that $|f(\mathbf{x}_k + \mathbf{s}) - f(\mathbf{x}_k) - \mathbf{g}_k^\top \mathbf{s} - \frac{1}{2}\mathbf{s}^\top \mathbf{H}_k \mathbf{s}| \le \frac{L_2}{6}\|\mathbf{s}\|^3$. Thus, by denoting $\rho^{(m)}(\mathbf{H}_k, \mathbf{g}_k)$ by $\rho_k^{(m)}$, together with (7) we can get

$$f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k + \mathbf{s}) + \frac{\rho_k^{(m)}}{m}\|\mathbf{s}\|^2 + \frac{L_2 + M}{6}\|\mathbf{s}\|^3, \quad \forall \mathbf{s} \in \mathbb{R}^d. \tag{24}$$

Now define two auxiliary sequences $\{a_k\}_{k\ge 0}$ and $\{A_k\}_{k \ge 0}$ by $a_k = k^2$, $A_0 = 0$ and $A_k = A_{k-1} + a_k$. By choosing $\mathbf{s} = \frac{a_{k+1}}{A_{k+1}}(\mathbf{x}^* - \mathbf{x}_k)$ in (24), we get

$$f(\mathbf{x}_{k+1}) \le f\left(\frac{A_k}{A_{k+1}}\mathbf{x}_k + \frac{a_{k+1}}{A_{k+1}}\mathbf{x}^*\right) + \frac{\rho_k^{(m)}}{m}\frac{a_{k+1}^2}{A_{k+1}^2}\|\mathbf{x}^* - \mathbf{x}_k\|^2 + \frac{L_2 + M}{6}\frac{a_{k+1}^3}{A_{k+1}^3}\|\mathbf{x}^* - \mathbf{x}_k\|^3 \tag{25}$$

$$\le \frac{A_k}{A_{k+1}} f(\mathbf{x}_k) + \frac{a_{k+1}}{A_{k+1}} f(\mathbf{x}^*) + \frac{\rho_k^{(m)}}{m}\frac{a_{k+1}^2}{A_{k+1}^2}\|\mathbf{x}^* - \mathbf{x}_k\|^2 + \frac{L_2 + M}{6}\frac{a_{k+1}^3}{A_{k+1}^3}\|\mathbf{x}^* - \mathbf{x}_k\|^3, \tag{26}$$

where we used Jensen's inequality in the last inequality. By multiplying both sides by A_{k+1} and using the upper bound $\|\mathbf{x}_k - \mathbf{x}^*\| \le D$, we obtain

$$A_{k+1}\left(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)\right) \le A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{\rho_k^{(m)}}{m} \cdot \frac{a_{k+1}^2}{A_{k+1}} D^2 + \frac{L_2 + M}{6} \cdot \frac{a_{k+1}^3}{A_{k+1}^2} D^3.$$

Note that $A_{k+1} = \sum_{j=1}^{k+1} a_j = \sum_{j=1}^{k+1} j^2 \ge (k+1)^3/3$ and $a_{k+1} = (k+1)^2$. Hence, we further have

$$A_{k+1}\left(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)\right) \le A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{3\rho_k^{(m)}}{m}(k+1)D^2 + \frac{3}{2}(L_2 + M)D^3.$$

By unrolling the above inequality, we obtain

$$A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) \le \frac{3D^2}{m}\sum_{j=0}^{k-1}(j+1)\rho_j^{(m)} + \frac{3}{2}(L_2 + M)D^3 k.$$

Finally, by using $\rho_j^{(m)} \le \rho_{\max}^{(m)}$ and $A_k \ge k^3/3$, this implies that

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \le \frac{9\rho_{\max}^{(m)} D^2}{2m}\left(\frac{1}{k} + \frac{1}{k^2}\right) + \frac{9(L_2 + M)D^3}{2k^2}.$$

C.4 Proof of Theorem 2

To simplify the notation, we define $T_M^{(m)} : \mathbb{R}^d \to \mathbb{R}^d$ as the operator that maps the current iterate $\mathbf{x}_k$ to the next iterate $\mathbf{x}_{k+1}$ in Algorithm 1. Formally, from (21) and (22) we have

$$T_M^{(m)}(\mathbf{x}) = \mathbf{x} + \operatorname*{arg\,min}_{\mathbf{s} \in \mathcal{K}_m(\nabla^2 f(\mathbf{x}), \nabla f(\mathbf{x}))} \left\{\langle \nabla f(\mathbf{x}), \mathbf{s}\rangle + \frac{1}{2}\mathbf{s}^\top \nabla^2 f(\mathbf{x})\mathbf{s} + \frac{M}{6}\|\mathbf{s}\|^3\right\}.$$

Thus, Algorithm 1 can be equivalently written as $\mathbf{x}_{k+1} = T_M^{(m)}(\mathbf{x}_k)$ for $k \ge 0$.

To prove Theorem 2, we artificially divide the iterates of Algorithm 1 into different stages and use $\mathbf{x}_{i,t}$ to denote the t-th iterate in the i-th stage. Also, the i-th stage has length k_i which we will specify later. Formally, in the first stage, we set $\mathbf{x}_{1,0} = \mathbf{x}_0$ and let $\mathbf{x}_{1,t+1} = T_M^{(m)}(\mathbf{x}_{1,t})$ for $0 \le t \le k_1 - 1$. Subsequently, in the i-th stage ($i \ge 2$), we set the initial iterate $\mathbf{x}_{i,0}$ as the last iterate of the previous stage $\mathbf{x}_{i-1,k_{i-1}}$ and let $\mathbf{x}_{i,t+1} = T_M^{(m)}(\mathbf{x}_{i,t})$ for $0 \le t \le k_i - 1$. Note that we have not altered the algorithm and the above procedure is exactly equivalent to Algorithm 1; The different stages are introduced solely for the purpose of analysis.

We choose the length k_i of the i-th stage such that the suboptimality gap halves after every stage, that is, $f(\mathbf{x}_{i+1,0}) - f(\mathbf{x}^*) \leq \frac{1}{2}(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*))$ for $i \geq 0$. In particular, we claim that it is sufficient to choose

$$k_i = \left\lceil \frac{72\rho_{\max}^{(m)}}{m\mu} + \frac{6 \cdot 2^{\frac{1}{4}}(L_2+M)^{\frac{1}{2}}\left(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*)\right)^{\frac{1}{4}}}{\mu^{\frac{3}{4}}} \right\rceil \leq \frac{72\rho_{\max}^{(m)}}{m\mu} + \frac{8(L_2+M)^{\frac{1}{2}}\left(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*)\right)^{\frac{1}{4}}}{\mu^{\frac{3}{4}}} + 1.$$

Let $D_i = \sup\{\|\mathbf{x} - \mathbf{x}^*\| : \ \mathbf{x} \in \mathbb{R}^d, \ f(\mathbf{x}) \leq f(\mathbf{x}_{i,0})\}$. Note that for any $\mathbf{x}$ satisfying $f(\mathbf{x}) \leq f(\mathbf{x}_{i,0})$, by using strong convexity we can bound $\frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^*\|^2 \leq f(\mathbf{x}) - f(\mathbf{x}^*) \leq f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*)$. This implies that $\|\mathbf{x} - \mathbf{x}^*\| \leq \sqrt{\frac{2(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*))}{\mu}}$ and thus we have $D_i \leq \sqrt{\frac{2(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*))}{\mu}}$. Hence, by applying Theorem 1, we can bound that

$$\begin{aligned} f(\mathbf{x}_{i,k_i}) - f(\mathbf{x}^*) &\leq \frac{9\rho_{\max}^{(m)} D_i^2}{2m}\left(\frac{1}{k_i} + \frac{1}{k_i^2}\right) + \frac{9(L_2+M)D_i^3}{2k_i^2} \\ &\leq \frac{9\rho_{\max}^{(m)}}{m\mu}\left(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*)\right)\left(\frac{1}{k_i} + \frac{1}{k_i^2}\right) + \frac{9\sqrt{2}(L_2+M)}{k_i^2}\left(\frac{f(\mathbf{x}_{k,0}) - f(\mathbf{x}^*)}{\mu}\right)^{\frac{3}{2}}. \end{aligned}$$

Since we choose k_i such that $k_i \geq \frac{72\rho_{\max}^{(m)}}{m\mu}$, we have

$$\frac{9\rho_{\max}^{(m)}}{m\mu}\left(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*)\right)\left(\frac{1}{k_i} + \frac{1}{k_i^2}\right) \leq \frac{18\rho_{\max}^{(m)}}{m\mu k_i}\left(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*)\right) \leq \frac{1}{4}\left(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*)\right).$$

Moreover, since $k_i \geq 6 \cdot 2^{\frac{1}{4}}\frac{(L_2+M)^{\frac{1}{2}}(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*))^{\frac{1}{4}}}{\mu^{\frac{3}{4}}}$, we also have

$$\frac{9\sqrt{2}(L_2+M)}{k_i^2}\left(\frac{f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*)}{\mu}\right)^{\frac{3}{2}} \leq \frac{1}{4}\left(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*)\right).$$

By combining the two, we conclude that $f(\mathbf{x}_{i+1,0}) - f(\mathbf{x}^*) = f(\mathbf{x}_{i,k_i}) - f(\mathbf{x}^*) \leq \frac{1}{2}\left(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*)\right)$. By induction, we have $f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*) \leq (f(\mathbf{x}_0) - f(\mathbf{x}^*))/2^{i-1} = \delta_0/2^{i-1}$. Hence, after at most $i^* = \lceil \log_2(\frac{\delta_0}{\epsilon}) + 1 \rceil$ stages, we have $f(\mathbf{x}_{i^*,0}) \leq \epsilon$. Furthermore, the total number of iterations can be bounded by

$$\begin{aligned} \sum_{i=1}^{i^*-1} k_i &\leq \frac{72\rho_{\max}^{(m)}}{m\mu}\left(\log_2\left(\frac{\delta_0}{\epsilon}\right) + 1\right) + \log_2\left(\frac{\delta_0}{\epsilon}\right) + 1 + \sum_{i=1}^{\infty}\frac{8(L_2+M)^{\frac{1}{2}}\left(f(\mathbf{x}_{i,0}) - f(\mathbf{x}^*)\right)^{\frac{1}{4}}}{\mu^{\frac{3}{4}}} \\ &\leq \frac{72\rho_{\max}^{(m)}}{m\mu}\left(\log_2\left(\frac{\delta_0}{\epsilon}\right) + 1\right) + \log_2\left(\frac{\delta_0}{\epsilon}\right) + 1 + \frac{8(L_2+M)^{\frac{1}{2}}\left(f(\mathbf{x}_0) - f(\mathbf{x}^*)\right)^{\frac{1}{4}}}{(1-2^{-\frac{1}{4}})\mu^{\frac{3}{4}}}. \end{aligned}$$

D UPPER BOUNDS ON $\rho^{(m)}$

In this section, we first present the proof of Lemma 4, which relates $\rho^{(m)}(\mathbf{A}, \mathbf{b})$ to matrix polynomials. Then in the subsequent sections, we will use this lemma to derive different bounds on $\rho^{(m)}(\mathbf{A}, \mathbf{b})$ by choosing different polynomials. As it will play an important role in our proof, we first briefly recap the definition of Chebyshev polynomials and present some useful properties (Saad, 2011, Section 4.4).

The Chebyshev polynomial of the first kind of degree k can be defined by

$$T_k(x) = \cos(k \cos^{-1} x).$$

Equivalently, it can also be defined via the following recurrence relation:

$$T_0(x) = 1, \ T_1(x) = x, \ T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x) \quad \forall k \geq 1.$$

For convenience, we list some useful properties of the Chebyshev polynomials in the following proposition.

Proposition 4. *The following statements hold true.*

(i) $|T_k(x)| \leq 1$ *for all* $x \in [-1, 1]$ *and* $k \geq 0$.

(ii) For $k \geq 1$, *the leading coefficient of* $T_k(x)$ *is* 2^{k-1}.

D.1 Proof of Lemma 4

By items (i) and (iii) in Proposition 8, we can write

$$\rho^{(m)}(\mathbf{A},\mathbf{b}) = \left(\prod_{j=1}^{m}\beta_{j+1}\right)^{\frac{1}{m}} = \min_{\mathbf{u}\in\mathcal{K}_m(\mathbf{A},\mathbf{b})} \|\mathbf{A}^m\mathbf{v}_1 - \mathbf{u}\|^{\frac{1}{m}},$$

where we recall that $\mathbf{v}_1 = \mathbf{b}/\|\mathbf{b}\|$. Moreover, for any $\mathbf{u}\in\mathcal{K}_m(\mathbf{A},\mathbf{b})$, by definition, there exist $c_0,\ldots,c_{m-1}\in\mathbb{R}$ such that

$$\mathbf{u} = \sum_{j=0}^{m-1} c_j\mathbf{A}^j\mathbf{b} = \left(\sum_{j=0}^{m-1} c_j\mathbf{A}^j\right)\mathbf{b} = \left(-\sum_{j=0}^{m-1}\tilde{c}_j\mathbf{A}^j\right)\frac{\mathbf{b}}{\|\mathbf{b}\|},$$

where we let $\tilde{c}_j = -c_j\|\mathbf{b}\|$. Hence, we have

$$\mathbf{A}^m\mathbf{v}_1 - \mathbf{u} = \left(\mathbf{A}^m + \sum_{j=0}^{m-1}\tilde{c}_j\mathbf{A}^j\right)\frac{\mathbf{b}}{\|\mathbf{b}\|} = \frac{p(\mathbf{A})\mathbf{b}}{\|\mathbf{b}\|},$$

where p is a monic polynomial of degree m as defined in Lemma 4. Thus, minimizing $\mathbf{u}$ over the subspace $\mathcal{K}_m(\mathbf{A},\mathbf{b})$ is equivalent to minimizing p over the set $\mathcal{M}_m$, which leads to

$$\rho^{(m)}(\mathbf{A},\mathbf{b}) = \min_{p\in\mathcal{M}_m}\left\|p(\mathbf{A})\frac{\mathbf{b}}{\|\mathbf{b}\|}\right\|^{\frac{1}{m}}.$$

This completes the proof.

D.2 Proof of Lemma 2

Assume that $\mathbf{A}$ has r distinct eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_r$, where $r \le d$. As discussed in Section 4.3, for any monic polynomial $\hat{p}\in\mathcal{M}_m$, by using Lemma 4 we can bound

$$\rho^{(m)}(\mathbf{A},\mathbf{b}) \le \min_{p\in\mathcal{M}_m}\|p(\mathbf{A})\|_{\mathrm{op}}^{\frac{1}{m}} \le \min_{p\in\mathcal{M}_m}\max_{i\in\{1,2,\ldots,r\}}|p(\lambda_i)|^{\frac{1}{m}} \le \max_{i\in\{1,2,\ldots,r\}}|\hat{p}(\lambda_i)|^{\frac{1}{m}}. \tag{27}$$

Since $0\preceq\mathbf{A}\preceq L_1\mathbf{I}$, we have $0\le\lambda_i\le L_1$ for all $i=1,\ldots,r$. Now we will choose the polynomial $\hat{p}$ as

$$\hat{p}(\lambda) = 2^{-m+1}\left(\frac{L_1}{2}\right)^m T_m\left(\frac{2\lambda}{L_1}-1\right) = 2\left(\frac{L_1}{4}\right)^m T_m\left(\frac{2\lambda}{L_1}-1\right).$$

By item (ii) in Proposition 4, we can verify that $\hat{p}$ is indeed a monic polynomial of degree m. Moreover, for any $\lambda\in[0,L_1]$, we can obtain from item (i) in Proposition 4 that $|\hat{p}(\lambda)|\le 2\left(L_1/4\right)^m$. hence, we conclude that $\rho^{(m)}(\mathbf{A},\mathbf{b})\le 2^{1/m}L_1/4$.

D.3 Proof of Lemma 5

Assume that $\mathbf{H}$ has r distinct eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_r$, where $r\le d$. We follow similar arguments as in Section D.2 but choose the polynomial $\hat{p}$ differently. Specifically, in the first case when $m < r$, we let

$$\hat{p}(\lambda) = \prod_{j=1}^{m}(\lambda-\lambda_j).$$

It is easy to verify that $\hat{p}$ is a monic polynomial of degree m with its zeros at $\lambda_1,\ldots,\lambda_m$. Thus, we immediately get $\hat{p}(\lambda_i)=0$ for $i\in\{1,2,\ldots,m\}$. On the other hand, for any $i\in\{m+1,m+2,\ldots,r\}$, we have $0\le\lambda_i\le\lambda_m$ and thus $\hat{p}(\lambda_i)\le\prod_{j=1}^m|\lambda_i-\lambda_j|\le\prod_{j=1}^m\lambda_j$. Hence, by (27) we conclude that $\rho^{(m)}(\mathbf{H},\mathbf{g})\le\left(\prod_{j=1}^m\lambda_j\right)^{\frac{1}{m}}$.

In the second case when $m\ge r$, we can let

$$\hat{p}(\lambda) = (\lambda-\lambda_1)^{m-r}\prod_{j=1}^{r}(\lambda-\lambda_j).$$

Algorithm 3 Cubic regularized Newton with line search

Input: Initial point $\mathbf{x}_0 \in \mathbb{R}^d$, initial regularization parameter $R_0 > 0$, line search parameter $\beta \in (0,1)$
for $k = 0, 1, \ldots,$ **do**
  Let M_k be the smallest number in $\{R_k \beta^{-i} : i \geq 0\}$ such that

$$\mathbf{s}_k = \arg\min_{\mathbf{s} \in \mathbb{R}^d} \Big\{ \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M_k}{6} \|\mathbf{s}\|^3 \Big\},$$
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k,$$
$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M_k}{6} \|\mathbf{s}\|^3.$$

  Set $R_{k+1} \leftarrow \beta M_k$
end for

We can observe that $\hat{p}$ is a monic polynomial of degree m and it vanishes at all λ_i for $i = 1, 2, \ldots, r$. Thus, we conclude that $\rho^{(m)}(\mathbf{H}, \mathbf{g}) = 0$.

Finally, we prove our claim in Remark 4. We first make a simple observation that $\mathbf{H}^m \mathbf{g} \in \mathcal{K}_m(\mathbf{H}, \mathbf{g})$ if and only if $m \geq r_0$, where we recall r_0 denotes the dimension of the maximal Krylov subspace. This is because, by definition, we have $\mathcal{K}_m(\mathbf{H}, \mathbf{g}) \subsetneq \mathcal{K}_{m+1}(\mathbf{H}, \mathbf{g})$ for $m < r_0$ and $\mathcal{K}_m(\mathbf{H}, \mathbf{g}) = \mathcal{K}_{m+1}(\mathbf{H}, \mathbf{g})$ for $m \geq r_0$. To complete the proof, Lemma 4 implies that $\rho^{(m)}(\mathbf{H}, \mathbf{g}) = 0$ if and only if there exists a polynomial $p(x) = x^m + \sum_{i=0}^{m-1} c_i x^i \in \mathcal{M}_m$ such that $p(\mathbf{H})\mathbf{g} = 0$, i.e., $\mathbf{H}^m \mathbf{g} = -\sum_{i=0}^{m-1} c_i \mathbf{H}^i \mathbf{g}$. Moreover, this is exactly equivalent to the condition that $\mathbf{H}^m \mathbf{g} \in \mathcal{K}_m(\mathbf{H}, \mathbf{g})$, and hence by our initial observation above we have $\rho^{(m)}(\mathbf{H}, \mathbf{g}) = 0$ if and only if $m \geq r_0$.

D.4 Proof of Lemma 6

Inspired by Goujaud et al. (2022), we choose the polynomial $\hat{p}$ as

$$\hat{p}(\lambda) = 2^{1-\frac{m}{2}} \left(\frac{\Delta(L_1 - \Delta)}{2} \right)^{\frac{m}{2}} T_{m/2}(\sigma(\lambda)), \quad \text{where } \sigma(\lambda) = 1 - \frac{2}{\Delta(L_1 - \Delta)} \lambda(L_1 - \lambda).$$

We can verify that this is a monic polynomial of degree m. Moreover, for any $\lambda \in [0, \Delta] \cup [L_1 - \Delta, L_1]$, it holds that $\sigma(\lambda) \in [-1, 1]$. Hence, using item (i) in Proposition 4, for any $i = 1, 2, \ldots, r$ we obtain that $|T_{m/2}(\sigma(\lambda_i))| \leq 1$, which further implies that $|\hat{p}(\lambda)| \leq 2^{1-\frac{m}{2}} \left(\frac{\Delta(L_1 - \Delta)}{2} \right)^{\frac{m}{2}}$. Thus, by (27) we get $\rho^{(m)}(\mathbf{H}, \mathbf{g}) \leq 2^{1/m} \sqrt{\Delta(L_1 - \Delta)}/2$.

E EXPERIMENT DETAILS

In the experiments, we focus on the logistic regression problem with LIBSVM datasets (Chang and Lin, 2011). Specifically, consider a dataset $\{(\mathbf{a}_j, b_j)\}_{j=1}^n$ of n points, where $\mathbf{a}_j \in \mathbb{R}^d$ is the j-th feature vector and $b_j \in \{0, 1\}$ is the j-th binary label. Then, the logistic loss function is given by $f(\mathbf{x}) = \frac{1}{n} \sum_{j=1}^n \Big((1 - b_j) \mathbf{a}_j^\top \mathbf{x} + \log(1 + e^{-\mathbf{a}_j^\top \mathbf{x}}) \Big)$. We tested the CRN method (Nesterov and Polyak, 2006), the SSCN method (Hanzely et al., 2020), and our proposed Krylov CRN method (Algorithm 1). In the following sections, we further describe their implementation details.

E.1 Cubic Regularized Newton

Recall that in the analysis of CRN, the regularization parameter M should satisfy $M \geq L_2$. Since the Lipschitz constant L_2 is unknown in practice, we use a backtracking line search scheme to select M_k at the k-th iteration, as described in Algorithm 3. Specifically, at the k-th iteration, we iteratively increase the value of the regularization parameter M_k until the key inequality in (4) is satisfied.

To solve the cubic subproblem in (2), we follow the standard approach in Cartis et al. (2011, Section 6.1). As we mentioned in Section 2.1, by using the first-order optimality condition, the cubic subproblem can be reformulated as the nonlinear equation:

$$\lambda = \frac{M}{2} \|(\mathbf{H}_k + \lambda \mathbf{I})^{-1} \mathbf{g}_k\|.$$

Algorithm 4 Stochastic Subspace Cubic Newton with line search

Input: Initial point $\mathbf{x}_0 \in \mathbb{R}^d$, subspace dimension m, initial regularization parameter $R_0 > 0$, line search parameter $\beta \in (0,1)$
for $k = 0, 1, \ldots,$ **do**
  Sample m coordinates uniformly and randomly and let I_k be the set of sampled indices
  Compute the subspace gradient $\tilde{\mathbf{g}}_k$ and the subspace Hessian $\tilde{\mathbf{H}}_k$
  Let M_k be the smallest number in $\{R_k\beta^{-i} : i \geq 0\}$ such that

$$\begin{aligned}\mathbf{z}_k &= \arg\min_{\mathbf{z}\in\mathbb{R}^m} \Big\{\tilde{\mathbf{g}}_k^\top \mathbf{z} + \frac{1}{2}\mathbf{z}^\top \tilde{\mathbf{H}}_k \mathbf{z} + \frac{M_k}{6}\|\mathbf{z}\|^3\Big\},\\ \mathbf{x}_{k+1}[I_k] &= \mathbf{x}_k[I_k] + \mathbf{z}_k,\\ f(\mathbf{x}_{k+1}) &\leq f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M_k}{6}\|\mathbf{s}\|^3.\end{aligned}$$

  Set $R_{k+1} \leftarrow \beta M_k$
end for

Algorithm 5 Krylov cubic regularized Newton with line search

Input: Initial point $\mathbf{x}_0 \in \mathbb{R}^d$, subspace dimension m, initial regularization parameter $R_0 > 0$, line search parameter $\beta \in (0,1)$
for $k = 0, 1, \ldots,$ **do**
  Set $(\mathbf{V}_k, \tilde{\mathbf{g}}_k, \tilde{\mathbf{H}}_k) = \textsc{Lanczos}(\mathbf{H}_k, \mathbf{g}_k; m)$
  Let M_k be the smallest number in $\{R_k\beta^{-i} : i \geq 0\}$ such that

$$\begin{aligned}\mathbf{z}_k &= \arg\min_{\mathbf{z}\in\mathbb{R}^m} \Big\{\tilde{\mathbf{g}}_k^\top \mathbf{z} + \frac{1}{2}\mathbf{z}^\top \tilde{\mathbf{H}}_k \mathbf{z} + \frac{M_k}{6}\|\mathbf{z}\|^3\Big\},\\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{V}_k \mathbf{z}_k,\\ f(\mathbf{x}_{k+1}) &\leq f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_k \mathbf{s} + \frac{M_k}{6}\|\mathbf{s}\|^3.\end{aligned}$$

  Set $R_{k+1} \leftarrow \beta M_k$
end for

Thus, we define the univariate function $\phi(\lambda) = \lambda^2 - \frac{M^2}{4}\|(\mathbf{H}_k + \lambda\mathbf{I})^{-1}\mathbf{g}_k\|^2$ and use the Newton's method to find its unique root λ^*. Then, we can compute the solution by $\mathbf{s}_k = -(\mathbf{H}_k + \lambda^*\mathbf{I})^{-1}\mathbf{g}_k$. Note that at each iteration of Newton's method, we need to solve a linear system of equations in the form of $(\mathbf{H}_k + \lambda\mathbf{I})\mathbf{s} = -\mathbf{g}_k$ with a different λ. When the problem dimension d is small (less than 500 in our experiment), we can store the Hessian matrix $\mathbf{H}_k$ and solve the linear system directly by computing the matrix inverse. On the other hand, when d is large, we need to use the conjugate gradient method to solve the linear system relying on Hessian-vector products.

E.2 Stochastic Subspace Cubic Newton

We implemented the coordinate version of SSCN, where we sample m coordinates uniformly and randomly at each iteration. We use a similar backtracking line search scheme in SSCN to select the regularization parameter M_k as shown in Algorithm 4. Given an index set I, we use $\mathbf{x}[I]$ to denote the subvector indexed by I. Moreover, to achieve its best performance, we follow the strategy in (Hanzely et al., 2020, Section 7.1) and store the residuals $\mathbf{a}_j^\top \mathbf{x}_k$ for each data point $j = 1, 2, \ldots, n$ and at each iteration k. As shown in Hanzely et al. (2020), in this case the computational cost of the subspace gradient and the subspace Hessian is $\mathcal{O}(nm)$ and $\mathcal{O}(nm^2)$, respectively.

E.3 Krylov Cubic Regularized Newton

To implement our proposed method, we also use a backtracking line search scheme to select the regularization parameter. The full algorithm is given in Algorithm 5.