Incremental Quasi-Newton Methods with Faster Superlinear Convergence Rates

Zhuanghua Liu[1, 2], **Luo Luo***[3], **Bryan Kian Hsiang Low**[1]

[1]Department of Computer Science, National University of Singapore
[2]CNRS@CREATE LTD, 1 Create Way, #08-01 CREATE Tower, Singapore 138602
[3]School of Data Science, Fudan University
liuzhuanghua9@gmail.com, luoluo@fudan.edu.cn, lowkh@comp.nus.edu.sg

Abstract

We consider the finite-sum optimization problem, where each component function is strongly convex and has Lipschitz continuous gradient and Hessian. The recently proposed incremental quasi-Newton method is based on BFGS update and achieves a local superlinear convergence rate that is dependent on the condition number of the problem. This paper proposes a more efficient quasi-Newton method by incorporating the symmetric rank-1 update into the incremental framework, which results in the condition-number-free local superlinear convergence rate. Furthermore, we can boost our method by applying the block update on the Hessian approximation, which leads to an even faster local convergence rate. The numerical experiments show the proposed methods significantly outperform the baseline methods.

1 Introduction

We study the following finite-sum minimization problem:

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

where each individual function $f_i : \mathbb{R}^d \to \mathbb{R}$ is strongly convex and has Lipschitz continuous gradient and Hessian. This formulation is ubiquitous in various machine learning models, including maximum likelihood estimation (MLE) (Bishop and Nasrabadi 2006; Bottou, Curtis, and Nocedal 2018) and unsupervised learning problems (Hastie et al. 2009; Murphy 2012). A notable example of the problem (1) is the empirical risk minimization in supervised learning, where n is the number of data examples and $f_i(\cdot)$ corresponds to the loss function incurred by each training instance.

In this paper, we are interested in solving the large-scale finite-sum problem, that is, the number of components n in formulation (1) is large. In this scenario, accessing the exact gradient or Hessian over the entire dataset is too expensive for each iteration. To circumvent this issue, stochastic or incremental optimization methods were introduced since they only require computing an estimation of the gradient or Hessian by a single sample (or a small mini-batch of samples) at each round. The most popular of these methods is stochastic gradient descent (SGD). It has been widely used in large-scale optimization problems thanks to its cheap computational cost per iteration (Bottou, Curtis, and Nocedal 2018). Applying the variance reduction (Defazio, Bach, and Lacoste-Julien 2014; Johnson and Zhang 2013; Schmidt, Le Roux, and Bach 2017; Zhang, Mahdavi, and Jin 2013) and acceleration techniques (Allen-Zhu 2017; Nesterov 2003) can improve the vanilla SGD, and it achieves a linear convergence rate with optimal incremental first-order oracle complexity (Woodworth and Srebro 2016).

The second-order methods (Nesterov 2003) incorporate the additional curvature information in every iteration, and it is possible to establish the local superliner convergence rate with these methods. For the finite-sum problem (1), Rodomanov and Kropotov (2016) proposed the Newton incremental method (NIM), which requires accessing the exact gradient and exact Hessian of one individual function in each iteration and attains the local superlinear convergence rate. The classical quasi-Newton methods (Broyden, Dennis Jr, and Moré 1973; Dennis and Moré 1974; Powell 1971) estimate the second-order information with first-order oracle calls and still hold the superlinear convergence rate. However, most of the stochastic variants for quasi-Newton methods (Lucchi, McWilliams, and Hofmann 2015; Moritz, Nishihara, and Jordan 2016) that employ gradient estimators only achieve linear convergence rates.

Mokhtari, Eisen, and Ribeiro (2018) proposed the Incremental quasi-Newton (IQN) method by using classical BFGS update (Broyden 1970; Fletcher 1970; Goldfarb 1970; Shanno 1970), which is the first superlinear convergent quasi-Newton method without exact second-order oracle call in each iteration. However, the best-known analysis of IQN (Mokhtari, Eisen, and Ribeiro 2018) only provided the asymptotic convergence result. Several follow-up works (Gao, Koppel, and Ribeiro 2020; Lahoti et al. 2023) attempted to characterize the convergence rate by fusing the greedy quasi-Newton update (Rodomanov and Nesterov 2021a) into the framework of IQN. Specifically, Lahoti et al. (2023) proposed sharpened lazy incremental quasi-Newton (SLIQN) by utilizing lazy propagation strategy and showed it achieves the superlinear convergence rate of $\mathcal{O}\big((1-d^{-1}\varkappa^{-1})^{\lceil t/n\rceil^2}\big)$, where $\varkappa$ is the condition number and t is the number of iterations. Gao, Koppel, and Ribeiro (2020) proposed the

*The corresponding author

Incremental Greedy BFGS (IGS) method with the same convergence rate as the SLIQN, but it requires more expensive per-iteration complexity.

In this work, we propose an efficient quasi-Newton method named the Lazy Incremental Symmetric Rank-1 (LISR-1) method for the finite-sum minimization problem. Our approach takes advantage of the well-known symmetric rank-1 (SR1) update to construct the Hessian estimator with sharper error bound than BFGS methods, and it also exploits the lazy propagation strategy to maintain a low per-iteration complexity. We show that LISR-1 achieves a local superlinear convergence rate of $\mathcal{O}\big((1-d^{-1})^{\lceil t/n\rceil^2}\big)$, shaving off the dependency on the condition number $\varkappa$ compared with the convergence rate achieved by SLIQN and IGS. Each iteration of LISR-1 requires only $O(1)$ incremental gradient/Hessian-vector oracle calls and $\mathcal{O}(d^2)$ flops in matrix operations, matching the existing IQN methods. Furthermore, we extend LISR-1 by making use of the symmetric rank-k update (Liu, Chen, and Luo 2023) to construct the more accurate Hessian estimator where $k < d$ is the rank of the update, resulting in the block IQN method called Lazy Incremental Symmetric Rank-k (LISR-k). It enjoys the local convergence rate up to $\mathcal{O}\big((1-kd^{-1})^{\lceil t/n\rceil^2}\big)$ with additional computational cost of $\mathcal{O}(kd^2)$ flops per-iteration. The numerical experiments on quadratic programming problems and the model of regularized logistic regression demonstrate significant improvements over baseline methods and confirm our theoretical findings.

Paper Organization In Section 2, we provide a literature review for quasi-Newton methods and their variants for finite-sum optimization problems. In Section 3, we formalize the notations and assumptions of our problem and introduce the background of the Broyden family update. In Section 4, we propose our LISR-1 method and provide its convergence analysis. In Section 5, we present the LISR-k method by incorporating the block-type update. In Section 6, we demonstrate the numerical experiments to show the improved efficiency of the proposed methods. Finally, we conclude this work in Section 7. All the proofs and more experimental results are deferred to the appendix.

2 Related Work

In this section, we review related work of quasi-Newton methods and their variants for large-scale optimization problems.

Classical Quasi-Newton Methods Past decades have witnessed extensive research progress on quasi-Newton methods. The main advantage of quasi-Newton methods is their capability to reach a superlinear convergence without computing the exact Hessian or its inverse. To estimate the second-order information, the classical quasi-Newton methods are based on the secant equation and the corresponding closeness criteria between successive Hessian estimations. The choice of closeness criteria leads to different types of quasi-Newton methods, including Broyden's method (Broyden 1965; Broyden, Dennis Jr, and Moré 1973; Gay 1979), the Davidon-Fletcher-Powell (DFP) method (Davidon 1991; Fletcher and Powell 1963), the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method (Broyden 1970; Fletcher 1970; Goldfarb 1970; Shanno 1970) and the symmetric rank-1 (SR1) method (Conn, Gould, and Toint 1991). The asymptotic superlinear convergence of quasi-Newton methods was established in the 1970s (Broyden, Dennis Jr, and Moré 1973; Dennis and Moré 1974; Dixon 1972a,b; Powell 1971), while the explicit superlinear rates of quasi-Newton methods were obtained only recently. Rodomanov and Nesterov (2021a) first proposed greedy quasi-Newton methods and gave its non-asymptotic superlinear convergence guarantees. Later, Lin, Ye, and Zhang (2022) provided a sharper analysis for these methods. After that, Jin and Mokhtari (2023); Rodomanov and Nesterov (2021b,c); Ye et al. (2021) established the explicit rates for the classical (secant equation-based) quasi-Newton methods.

Block Quasi-Newton Methods Schnabel (1983) proposed block quasi-Newton methods. These methods construct the Hessian estimator along multiple directions during each iteration, and they achieve better empirical performance than classical quasi-Newton methods like BFGS (O'Leary and Yeremin 1994). After several decades, the superlinear convergence of these methods was established by Gao and Goldfarb (2018); Gower, Goldfarb, and Richtárik (2016); Gower and Richtárik (2017). Very recently, Liu, Chen, and Luo (2023) presented explicit superlinear convergence rates of block quasi-Newton methods, which explains why the use of multiple directions benefits the convergence behaviors.

Stochastic/Incremental Quasi-Newton Methods Due to the sheer volume of data in modern machine learning applications, researchers have been investigating the extension of quasi-Newton methods on large-scale optimization problems. Several early works established the stochastic quasi-Newton methods to reduce the computational cost at each iteration (Byrd et al. 2016; Chang, Sun, and Zhang 2019; Lucchi, McWilliams, and Hofmann 2015; Mokhtari and Ribeiro 2014, 2015; Moritz, Nishihara, and Jordan 2016), but these methods cannot obtain the superlinear convergences like classical quasi-Newton methods. Incremental quasi-Newton methods (IQN) (Gao, Koppel, and Ribeiro 2020; Lahoti et al. 2023; Mokhtari, Eisen, and Ribeiro 2018) use the aggregated information to construct a more accurate gradient and Hessian estimator, which leads to superlinear convergence. We compare the proposed methods with related work in Table 1.

3 Preliminaries

In this section, we formalize the notations and assumptions throughout this paper, then we introduce the well-known Broyden family updates which are widely used in quasi-Newton methods.

3.1 Notations

We denote $e_i \in \mathbb{R}^d$ as the i-th standard basis vector of d-dimensional Euclidean space, where $i \in [d]$. We define the index i_t as $t \bmod n$. For vectors $u, v \in \mathbb{R}^d$, we denote their inner product by $\langle u, v\rangle := u^\top v$. We use $\|\cdot\|$ to represent the Euclidean norm of the vector and the spectral norm of the matrix. Given a positive semi-definite matrix $A \in \mathbb{R}^{d\times d}$ and a vector $u \in \mathbb{R}^d$, we define the norm of u with respect to A

Algorithm	Computation Cost	Convergence Rate
IQN (Mokhtari, Eisen, and Ribeiro 2018)	$\mathcal{O}(d^2)$	asymptotic superlinear
IGS (Gao, Koppel, and Ribeiro 2020)	$\mathcal{O}(d^3)$	$\mathcal{O}\big((1-d^{-1}\varkappa^{-1})^{\lceil t/n\rceil^2}\big)$
SLIQN (Lahoti et al. 2023)	$\mathcal{O}(d^2)$	$\mathcal{O}\big((1-d^{-1}\varkappa^{-1})^{\lceil t/n\rceil^2}\big)$
LISR-1 (this work)	$\mathcal{O}(d^2)$	$\mathcal{O}\big((1-d^{-1})^{\lceil t/n\rceil^2}\big)$
LISR-k (this work)	$\mathcal{O}(kd^2)$	$\mathcal{O}\big((1-kd^{-1})^{\lceil t/n\rceil^2}\big)$

Table 1: We compare the per-iteration computation cost and the convergence rates of incremental fashion quasi-Newton methods. Note that the explicit convergence rate of the vanilla IQN method still remains a mystery.

as $\|u\|_A \coloneqq \sqrt{\langle u, Au\rangle}$. We let

$$E_k(A) = [e_{i_1}; \ldots; e_{i_k}] \in \mathbb{R}^{d\times k}, \tag{2}$$

where $i_1, \ldots, i_k$ are the indices for the largest k entries in the diagonal of A. We also use $\mathrm{tr}(\cdot)$ to present the trace of a square matrix. Additionally, we denote the solution of problem (1) as $x^* \coloneqq \arg\min_{x\in\mathbb{R}^d} f(x)$.

3.2 Assumptions

In the remainder of this paper, we always suppose Problem (1) satisfies the following assumptions.

Assumption 3.1. We suppose each function $f_i(\cdot)$ is twice-differentiable, L-smooth and μ-strongly convex, i.e., there exist constants $L > 0$ and $\mu > 0$ such that

$$\mu I \preceq \nabla^2 f_i(x) \preceq LI \tag{3}$$

for any $x \in \mathbb{R}^d$.

Assumption 3.2. We suppose each $f_i(\cdot)$ has a $\tilde{L}$-Lipschitz continuous Hessian, i.e., there exists a constant $\tilde{L}$ such that

$$\left\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\right\| \le \tilde{L}\left\|x - y\right\|.$$

for any $x, y \in \mathbb{R}^d$.

The strong convexity and the Lipschitz continuity of Hessian in our assumptions imply that each $f_i(\cdot)$ is strongly self-concordant with constant $M \coloneqq \tilde{L}\mu^{-3/2}$ (Rodomanov and Nesterov 2021a), i.e, we have

$$\nabla^2 f_i(y) - \nabla^2 f_i(x) \preceq M\left\|y - x\right\|_{\nabla^2 f_i(z)} \nabla^2 f_i(w)$$

for any $x, y, z, w \in \mathbb{R}^d$.

Additionally, we let $\varkappa \coloneqq L/\mu$ be the condition number of our problem which could be very large in practice.

3.3 Broyden Family Update

Many popular quasi-Newton methods such as DFP, BFGS, and SR1 belong to the Broyden family update (Nocedal and Wright 1999, Section 6.3), which is defined as follows.

Definition 3.3. Let $G \in \mathbb{R}^{d\times d}$ and $A \in \mathbb{R}^{d\times d}$ be two positive define matrices satisfying $G \succeq A$. For any non-zero $u \in \mathbb{R}^d$ and $\tau \in [0, 1]$, if $Gu = Au$, we define $\mathrm{Broyd}_\tau(G, A, u) \coloneqq G$. Otherwise, we define

$$\begin{aligned}&\mathrm{Broyd}_\tau(G, A, u)\\ \coloneqq&\tau\left[G - \frac{Auu^\top G + Guu^\top A}{u^\top Au} + \left(\frac{u^\top Gu}{u^\top Au} + 1\right)\frac{Auu^\top A}{u^\top Au}\right]\\ &+ (1-\tau)\left[G - \frac{(G-A)uu^\top(G-A)}{u^\top(G-A)u}\right].\end{aligned} \tag{4}$$

We can recover several well-known quasi-Newton methods by taking the different values of τ:

- For $\tau = 1$, Eq. (4) corresponds to the DFP update

$$\begin{aligned}\mathrm{DFP}(G, A, u) \coloneqq& G - \frac{Auu^\top G + Guu^\top A}{u^\top Au}\\ &+ \left(\frac{u^\top Gu}{u^\top Au} + 1\right)\frac{Auu^\top A}{u^\top Au}.\end{aligned}$$

- For $\tau = \dfrac{u^\top Au}{u^\top Gu}$, we recover the BFGS update

$$\mathrm{BFGS}(G, A, u) \coloneqq G - \frac{Guu^\top G}{u^\top Gu} + \frac{Auu^\top A}{u^\top Au}.$$

- For $\tau = 0$, we achieve the SR1 update

$$\mathrm{SR1}(G, A, u) \coloneqq G - \frac{(G-A)uu^\top(G-A)}{u^\top(G-A)u}. \tag{5}$$

We can generalize the Broyden family updates with multiple directions (Gao and Goldfarb 2018; Gower, Goldfarb, and Richtárik 2016; Gower and Richtárik 2017; Liu, Chen, and Luo 2023). In particular, Liu, Chen, and Luo (2023) establish the block version of the SR1 update called the symmetric rank-k (SR-k) update, which is defined as follows.

Definition 3.4. Let $A \in \mathbb{R}^{d\times d}$ and $G \in \mathbb{R}^{d\times d}$ be two positive-definite matrices satisfying $G \succeq A$. For any full rank matrix $U \in \mathbb{R}^{d\times k}$ with $k < d$, we define SR-$k(G, A, U) \coloneqq G$ if $GU = AU$. Otherwise, we define

$$\text{SR-}k(G, A, U) \coloneqq G - (G-A)U(U^\top(G-A)U)^\dagger U^\top(G-A).$$

Remark 3.5. Note that the SR-k update shown in the above definition is equivalent to the SR1 update when $k = 1$.

Algorithm 1: LISR-1

1: **Input**: $x^0 \in \mathbb{R}^d$ and $\{B_i^0 \in \mathbb{R}^{d\times d}\}_{i=1}^n$.
2: Initialize $t = 0$ and $z_i^0 = x^0$ for any $i \in [n]$.
3: **while** not converged **do**
4: Update x^{t+1} as per (6).
5: Update z_i^{t+1} as per (7).
6: Update B_i^{t+1} as per (8)-(11).
7: Increment the iteration counter t.
8: **end while**
9: **Output**: x^t.

4 Methodology

In this section, we propose the lazy incremental symmetric rank-1 (LISR-1) method and provide theoretical analysis to show it enjoys condition number-free local superlinear convergence.

4.1 The Algorithm

We first introduce the main intuitions of LISR-1. For each component function $f_i(x)$, we consider its quadratic approximation at point $z_i^t \in \mathbb{R}^d$ as

$$\begin{aligned} f_i(x) \approx & \tilde{f}_i^t(x) \\ = & f_i(z_i^t) + \nabla f_i(z_i^t)^\top (x - z_i^t) + \frac{1}{2}(x - z_i^t)^\top B_i^t (x - z_i^t), \end{aligned}$$

where we estimate $\nabla^2 f_i(z_i^t)$ by a positive-definite matrix $B_i^t \in \mathbb{R}^{d\times d}$. Then we obtain x^{t+1} by minimizing the average of $\{\tilde{f}_i^t(x)\}_{i=1}^n$, which has the closed form solution

$$\begin{aligned} x^{t+1} = & \arg\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \tilde{f}_i^t(x) \\ = & \left(\sum_{i=1}^n B_i^t\right)^{-1} \left(\sum_{i=1}^n B_i^t z_i^t - \sum_{i=1}^n \nabla f_i(z_i^t)\right). \end{aligned} \tag{6}$$

We only update one of $\{z_i^t\}_{i=1}^n$ at each iteration in a cyclic fashion to make the algorithm efficient, that is

$$z_i^{t+1} = \begin{cases} x^{t+1}, & \text{if } i = i_t, \\ z_i^t, & \text{otherwise}, \end{cases} \tag{7}$$

where $i_t = t \bmod n$ is the index of the component we choose at the t-th iteration.

We also wish to construct the Hessian estimators efficiently and keep a fast convergence rate. In particular, we introduce the scaling parameter ω^{t+1} and apply the SR1 update on one of the individual Hessian estimators in each iteration:

- For $i = i_t$, we let

$$B_i^{t+1} = \omega^{t+1}\mathrm{SR1}(B_i^t, \nabla^2 f_i(z_i^{t+1}), \bar{u}(B_i^t, \nabla^2 f_i(z_i^{t+1}))), \tag{8}$$

where $\bar{u}(\cdot,\cdot)$ is the greedy direction which is defined as

$$\bar{u}(G, A) \coloneqq \arg\max_{u\in\{e_i\}_{i=1}^d} u^\top (G - A)u. \tag{9}$$

- For $i \neq i_t$, we let

$$B_i^{t+1} = \omega^{t+1} B_i^t. \tag{10}$$

Additionally, we set

$$\omega^t = \begin{cases} \left(1 + M\sqrt{L} r_0 \rho^{\lceil \frac{t}{n} \rceil}\right)^2, & \text{if } n \bmod t = 0, \\ 1, & \text{otherwise}, \end{cases} \tag{11}$$

for some $\rho \in (0, 1 - d^{-1})$ and let r_0 be an upper bound of $\|x_0 - x^*\|$. This setting implies the step of scaling is executed once every n iterations.

We present the whole procedure of the proposed LISR-1 in Algorithm 1. We can verify that the per-iteration cost of our algorithm is $\mathcal{O}(d^2)$ flops. Notice that the main cost of LISR-1 comes from the computation of Eq. (6), which is dominated by maintaining the inverse of the following sum of individual Hessian estimators

$$\bar{B}^{t+1} \coloneqq \sum_{i=1}^n B_i^{t+1}.$$

We can rewrite the above matrix in the recursive form as

$$\bar{B}^{t+1} = \bar{B}^t + B_{i_t}^{t+1} - B_{i_t}^t. \tag{12}$$

In the case of $t \bmod n \neq 0$, no scaling is performed since we have $\omega^{t+1} = 1$. Denote $\bar{u}^t$ as the abbreviation of $\bar{u}(B_{i_t}^t, \nabla^2 f_{i_t}(z_{i_t}^t))$, then applying the Sherman-Morrison formula on Eq. (12) implies

$$(\bar{B}^{t+1})^{-1} = (\bar{B}^t)^{-1} + \frac{(\bar{B}^t)^{-1} v^t (v^t)^\top (\bar{B}^t)^{-1}}{(\bar{u}^t)^\top v^t - (v^t)^\top (\bar{B}^t)^{-1} v^t}, \tag{13}$$

where v^t is defined as

$$v^t = (B_{i_t}^t - \nabla^2 f_{i_t}(z_{i_t}^{t+1}))\bar{u}^t.$$

It is easy to observe that computing the right-hand side of Eq. (13) takes $\mathcal{O}(d^2)$ flops for given $(\bar{B}^t)^{-1}$ and v^t. In the case of $t \bmod n = 0$, each Hessian estimator may be scaled by a factor $\omega^t \neq 1$, which results in the additional computational cost of $\mathcal{O}(nd^2)$ flops. However, the amortized per-iteration complexity of this step is still $\mathcal{O}(d^2)$ because the scaling occurs once per n iterations. We provide a more efficient implementation of LISR-1 in the appendix.

4.2 Convergence Analysis

We analyze the convergence of LISR-1 by considering the Euclidean distance to the optimal solution x^*. Firstly, the formula (6) indicates the general result:

Lemma 4.1. *The iteration formula (6) satisfies*

$$\begin{aligned} \left\|x^{t+1} - x^*\right\| \leq & \frac{\tilde{L}\Gamma^t}{2} \sum_{i=1}^n \left\|z_i^t - x^*\right\|^2 \\ & + \Gamma^t \sum_{i=1}^n \left\|B_i^t - \nabla^2 f_i(z_i^t)\right\| \left\|z_i^t - x^*\right\|, \end{aligned} \tag{14}$$

for all $t \geq 1$, *where* $\Gamma^t \coloneqq \left\|\left(\sum_{i=1}^n B_i^t\right)^{-1}\right\|$.

Remark 4.2. Notice that the proof of Lemma 4.1 only requires the Lipschitz continuity of each $\nabla^2 f_i(\cdot)$ and the iteration formula (6). The validity of this lemma does not rely on the specific choice of Hessian estimators $\{B_i^t\}_{i=1}^n$ and it also can be used to analyze the other incremental fashion methods (Gao, Koppel, and Ribeiro 2020; Lahoti et al. 2023; Mokhtari, Eisen, and Ribeiro 2018).

In view of Lemma 4.1, the more accurate Hessian estimator $B_i^t \approx \nabla^2 f_i(z_i^t)$ can lead to the tighter upper bound of $\left\|x^{t+1} - x^*\right\|$. Hence, the key to showing the advantage of the proposed method is bounding the difference between B_i^t and $\nabla^2 f_i(z_i^t)$. In particular, we introduce the quantity

$$\nu(G, A) \coloneqq \frac{d \operatorname{tr}(G - A)}{\operatorname{tr}(A)}, \tag{15}$$

to describe the difference between two positive definite matrices $G \in \mathbb{R}^{d\times d}$ and $A \in \mathbb{R}^{d\times d}$ such that $G \succeq A$. Based on the measure $\nu(\cdot, \cdot)$ and Lemma 4.1, we provide the linear convergence of the distance to solution and the error of Hessian approximation as follows.

Lemma 4.3. *For any ρ satisfying $\rho \in (0, 1 - d^{-1})$, there exist positive constants r_0 and σ_0 such that running LISR-1 (Algorithm 1) with the initial conditions $\left\|x^0 - x^*\right\| \leq r_0$, $B_i^0 \succeq \omega^0 \nabla^2 f_i(z_i^0)$ and $\nu((\omega^0)^{-1} B_i^0, \nabla^2 f_i(x^0)) \leq \sigma_0$ for any $i = 1, \ldots, n$ results in*

$$\left\|x^{t+1} - x^*\right\| \leq \rho^{\lceil \frac{t+1}{n} \rceil} \left\|x^0 - x^*\right\| \tag{16}$$

and

$$\nu\big((\omega^{t+1})^{-1} B_{i_t}^{t+1}, \nabla^2 f_{i_t}(z_{i_t}^{t+1})\big) \leq \left(1 - \frac{1}{d}\right)^{\lceil \frac{t+1}{n} \rceil} \delta, \tag{17}$$

where $M = \tilde{L}/\mu^{3/2}$, ω_t follows the definition in (11) and

$$\delta \coloneqq \left(\sigma_0 + \frac{4MdL^{3/2}\mu^{-1} r_0}{1 - (1 - d^{-1})^{-1}\rho}\right) \exp\left(\frac{4M\sqrt{L} r_0}{1 - \rho}\right)$$

Remark 4.4. In the proof of Lemma 4.3, we show that the relation $(\omega^{t+1})^{-1} B_{i_t}^{t+1} \succeq \nabla^2 f_{i_t}(z_{i_t}^{t+1})$ holds for each iteration. This guarantees the update rule (6) and the error measure $\nu\big((\omega^{t+1})^{-1} B_{i_t}^{t+1}, \nabla^2 f_{i_t}(z_{i_t}^{t+1})\big)$ are well-defined.

We establish the mean-superlinear convergence based on Lemma 4.3. Specifically, we have the following result.

Lemma 4.5. *Following the initial conditions of Lemma 4.3, the sequence of iterates generated by the LISR-1 method (Algorithm 1) satisfies*

$$\left\|x^{t+1} - x^*\right\| \leq \left(1 - \frac{1}{d}\right)^{\lceil \frac{t+1}{n} \rceil} \cdot \frac{1}{n} \sum_{i=1}^{n} \left\|x^{t+1-i} - x^*\right\|.$$

Using Lemma 4.5, we can achieve the local superlinear convergence rate of the proposed LISR-1 method by induction. We formally present our main result as follows.

Theorem 4.6. *For the sequence $\{x_t\}$ generated by LISR-1 (Algorithm 1) with the initial conditions shown in Lemma 4.3, there exists a sequence $\{\zeta^l\}$ such that $\left\|x^{t+1} - x^*\right\| \leq \zeta^{\lfloor t/n \rfloor}$ for any $t \geq 1$ and it satisfies*

$$\zeta^l \leq r_0 \left(1 - \frac{1}{d}\right)^{\frac{(l+2)(l+1)}{2}}. \tag{18}$$

4.3 Discussion

The convergence analysis in the last subsection shows that LISR-1 enjoys the condition number-free superlinear convergence rate, which is significantly better than all of the existing incremental fashion quasi-Newton methods (see Table 1). The improvement is due to that we adopt the greedy SR1 update to maintain the Hessian estimator in formula (8) and the analysis characterizes the Hessian approximation error by the measure $\nu(\cdot, \cdot)$ defined in (15). In contrast, the prior methods IGS (Gao, Koppel, and Ribeiro 2020) and SLIQN (Lahoti et al. 2023) only consider the general Broyden family update and characterize the Hessian approximation error by the measure $\sigma(G, A) = \operatorname{tr}(A^{-1}(G - A))$ for positive definite $G \in \mathbb{R}^{d\times d}$ and $A \in \mathbb{R}^{d\times d}$, which leads to additional dependency on condition number in the superlinear convergence rate.[1] On the other hand, the implementations of these methods are more complicated than ours. Concretely, IGS requires scaling a Hessian estimator at each iteration which results in $\mathcal{O}(d^3)$ computational cost, and SLIQN maintains $B_{i_t}^{t+1}$ by a combination of secant equation-based and greedy Broyden family updates while our LISR-1 only has one step of greedy SR1 update (8).

5 Extension to Block Quasi-Newton Methods

It is possible to incorporate the idea of block quasi-Newton methods into the framework of the LISR-1. Specifically, we only need to modify Line 6 of Algorithm 1 by replacing the update rule (8) with

$$B_{i_t}^{t+1} = \omega^{t+1} \text{SR-}k(B_{i_t}^t, \nabla^2 f_{i_t}(z_{i_t}^{t+1}), \bar{U}(B_{i_t}^t, \nabla^2 f_{i_t}(z_{i_t}^{t+1}))), \tag{19}$$

where $\bar{U}(\cdot, \cdot)$ contains greedy directions which is defined as

$$\bar{U}(G, A) = E_k(G - A). \tag{20}$$

We name the variant of LISR-1 with the above modification as Lazy Incremental Symmetric Rank-k (LISR-k) method.

The LISR-k method requires $\mathcal{O}(kd^2)$ flops in each iteration. Since we typically set k to be much smaller than d, such computational cost is acceptable. Similar to the previous analysis, the cost of LISR-k is dominated by maintaining the inverse of the sum of individual Hessian estimators

$$\bar{B}^{t+1} \coloneqq \sum_{i=1}^{n} B_i^{t+1},$$

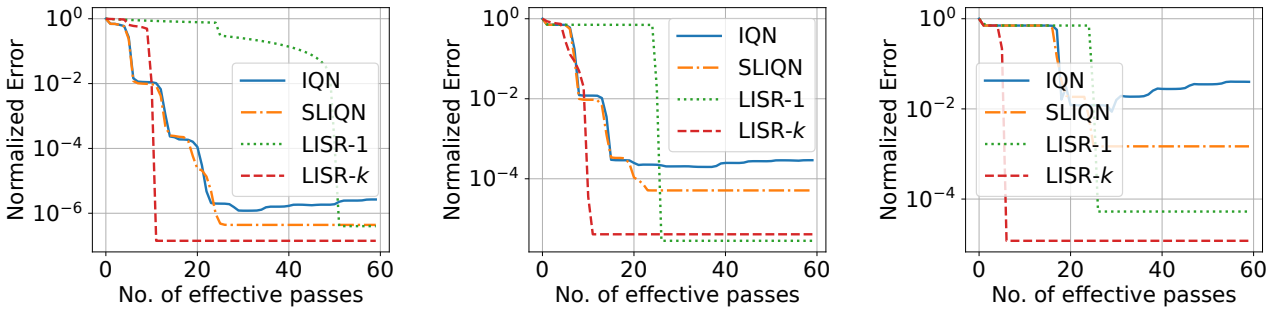
which can be written as $\bar{B}^t + B_{i_t}^{t+1} - B_{i_t}^t$. The main difference between the two algorithms is the update on $\bar{B}^{t+1}$ (its inverse) in the case of $t \bmod n \neq 0$. For the LISR-k method, we have

$$\bar{B}^{t+1} = \bar{B}^t - V^t \left((\bar{U}^t)^\top V^t\right)^{-1} (V^t)^\top,$$

where we define $V^t = \left(B_{i_t}^t - \nabla^2 f_{i_t}(z_{i_t}^{t+1})\right) \bar{U}^t \in \mathbb{R}^{d\times k}$ and $\bar{U}^t = \bar{U}(B_{i_t}^t, \nabla^2 f_{i_t}(z_{i_t}^{t+1})) \in \mathbb{R}^{d\times k}$. Applying the Sherman-Morrison formula, we achieve

$$(\bar{B}^{t+1})^{-1} = (\bar{B}^t)^{-1} + (\bar{B}^t)^{-1} V^t (D^t)^{-1} (V^t)^\top (\bar{B}^t)^{-1}, \tag{21}$$

[1] These work present their theoretical results by analyzing BFGS update, while their analysis can be directly applied to the general Broyden family update and achieves the identical convergence rate.



(a) $\xi = 4$, $\varkappa = 3.03 \times 10^2$ (b) $\xi = 8$, $\varkappa = 3.12 \times 10^4$ (c) $\xi = 12$, $\varkappa = 3.12 \times 10^6$

Figure 1: Normalized error vs. the number of effective passes for the quadratic programming problem.

where $D^t = (\bar{U}^t)^\top V^t - (V^t)^\top (\bar{B}^t)^{-1} V^t \in \mathbb{R}^{k \times k}$. It can be observed that constructing D^t takes $\mathcal{O}(kd^2)$ flops for given $(\bar{B}_t)^{-1}$. Additionally, the complexity of computing $(D^t)^{-1}$ is not the leading cost since we take $k \ll d$. Hence, the total cost for computing Eq. (21) is $\mathcal{O}(kd^2)$ flops. Similar to LISR-1, the setting of ω^{t+1} guarantees the scaling occurs once every n iterations and its amortized per-iteration complexity is no more than $\mathcal{O}(kd^2)$ flops.

Even Faster Convergence Rate The rank-k update in the LISR-k leads to sharper upper bounds on the distance to optimal solution and approximation error of Hessian estimators. Compared with Lemma 4.5, the LISR-k holds the following tighter upper bounds

$$\left\|x^{t+1} - x^*\right\| \leq \left(1 - \frac{k}{d}\right)^{\lceil \frac{t+1}{n} \rceil} \frac{1}{n} \sum_{i=1}^{n} \left\|x^{t+1-i} - x^*\right\|$$

and

$$\nu((\omega^{t+1})^{-1} B_{i_t}^{t+1}, \nabla^2 f_{i_t}(z_{i_t}^{t+1})) \leq \left(1 - \frac{k}{d}\right)^{\lceil \frac{t+1}{n} \rceil} \delta.$$

Consequently, we can show the mean-superlinear convergence result like Lemma 4.5, and the new result improves the base of convergence rate from $1 - d^{-1}$ to $1 - kd^{-1}$. Finally, we achieve the main result of the LISR-k method as follows.

Theorem 5.1. *We follow the initial conditions of Lemma 4.3 but initialize ρ with $\rho \in (0, 1 - kd^{-1})$. For the sequence of iterates $\{x^t\}$ generated by the LISR-k method, there exists sequence $\{\zeta^l\}$ such that $\|x^t - x^*\| \leq \zeta^{\lfloor (t-1)/n \rfloor}$ for any $t \geq 1$ and it satisfies*

$$\zeta^l \leq r_0 \left(1 - \frac{k}{d}\right)^{\frac{(l+2)(l+1)}{2}}. \tag{22}$$

Remark 5.2. For $k = 1$, the LISR-k method degenerates to the LISR-1 method. For $k \geq 2$, the superlinear convergence rate of LISR-k (Theorem 5.1) is strictly tighter than the counterpart of LISR-1 (Theorem 4.6).

6 Experiments

We compare the proposed methods LISR-1 and LISR-k with baseline methods including IQN (Mokhtari, Eisen, and Ribeiro 2018) and SLIQN (Lahoti et al. 2023). We test all methods on the problems of quadratic programming and regularized logistic regression. For the LISR-k method, we set $k = 5$ for all of the cases. For the fairness of comparison, we run all algorithms from the same initial point.

6.1 Quadratic Function Minimization

We consider the following quadratic function minimization problem

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{2}\langle x, A_i x\rangle + \langle b_i, x\rangle\right), \tag{23}$$

where $A_i \in \mathbb{R}^{d \times d}$ is positive definite and $b_i \in \mathbb{R}^d$. Following the setup of Mokhtari, Eisen, and Ribeiro (2018), we let each A_i be diagonal matrix by setting the first half of diagonal entries be independent uniformly sampled from $[1, 10^{\xi/2}]$ while the others are independent uniformly sampled from $[10^{-\xi/2}, 1]$, where $\xi > 0$ is the parameter that affects the condition number of the problem. For each b_i, we let its entries be independently uniformly sampled from $[0, 10^3]$.

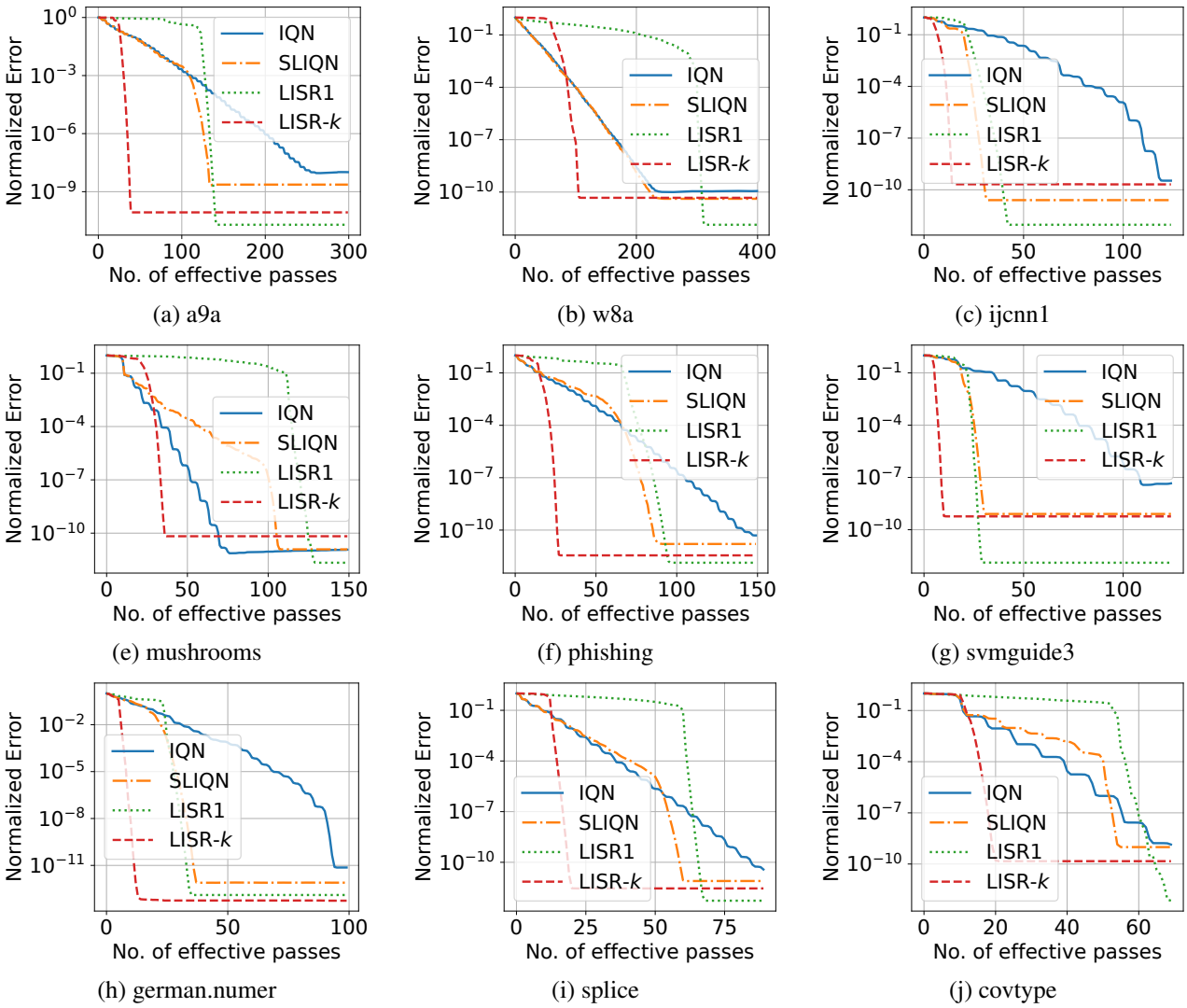
We run the experiments by taking $n = 1000$, $d = 50$ and $\xi \in \{4, 8, 12\}$, and we present the results in Figure 1. We observe that the condition number heavily affects the convergence behaviors of IQN and SLIQN, while the proposed methods LISR-1 and LISR-k are insensitive to the varying condition numbers. These results validate our theoretical analysis since we have shown the superlinear convergence rates of our methods do not depend on the condition number.

6.2 Regularized Logistic Regression

We consider ℓ_2-regularized logistic regression problem

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{i=1}^{n} \ln(1 + \exp(-y_i \langle x, z_i\rangle)) + \frac{\lambda}{2}\|x\|^2, \tag{24}$$

where $z_i \in \mathbb{R}^d$ is the feature of the i-th training sample and $y_i \in \{1, -1\}$ is the corresponding labels. We conduct our experiments on nine real-world datasets ("a9a", "w8a", "ijcnn", "mushrooms", "phishing", "svmguide3", "german.numer", "splice" and 'covtype") from LIBSVM repository. We take $\lambda = 10^{-3}$ for "a9a", "mushrooms", "svmguide3", "german.numer", "covtype" and $\lambda = 10^{-4}$ for others.



(a) a9a (b) w8a (c) ijcnn1

(e) mushrooms (f) phishing (g) svmguide3

(h) german.numer (i) splice (j) covtype

Figure 2: Normalized error vs. the number of effective passes for the regularized logistic regression problem on several real-world datasets .

We present the experimental results in Figure 2. We observe that the proposed LISR-k significantly outperforms other methods on all datasets. The LISR-1 enjoys a faster convergence rate than IQN and SLIQN when it starts to converge, while it may be slower at the early stage. We conjecture that IQN and SLIQN contain the steps of classical quasi-Newton updates. By accessing the exact gradient information, Rodomanov and Nesterov (2021b,c) theoretically showed that classical quasi-Newton methods converge faster than greedy quasi-Newton methods at the early stage. We empirically observe similar results for incremental quasi-Newton methods, while the rigorous theory for such a phenomenon is still unclear. On the other hand, the block update in LISR-k leads to much better Hessian estimators. Hence, the early stage of LISR-k only contains a few iterations.

7 Conclusion

This paper has proposed the efficient incremental quasi-Newton method called LISR-1 and its extension named LISR-k method for the finite-sum convex optimization. We have theoretically shown the proposed methods enjoy faster superlinear convergence rates than the state-of-the-art incremental quasi-Newton methods. The numerical experiments on quadratic programming and regularized logistic regression also validate the advantages of the proposed methods over existing IQN baselines.

In future work, it is interesting to study incremental quasi-Newton methods for more general settings, such as minimizing nonconvex functions (Wang et al. 2017; Yang et al. 2021). It is also possible to leverage the idea to design efficient incremental quasi-Newton methods for solving minimax problems (Liu et al. 2022; Liu and Luo 2022) or nonlinear equations (Liu et al. 2023).

Acknowledgments

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2023-08-043T-J). This research is part of the programme DesCartes and is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. Luo Luo is supported by National Natural Science Foundation of China (No. 62206058) and Shanghai Sailing Program (22YF1402900).

References

Allen-Zhu, Z. 2017. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, 1200–1205.

Bishop, C. M.; and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

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

Broyden, C. G. 1965. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92): 577–593.

Broyden, C. G. 1970. The convergence of single-rank quasi-Newton methods. *Mathematics of Computation*, 24(110): 365–382.

Broyden, C. G.; Dennis Jr, J. E.; and Moré, J. J. 1973. On the local and superlinear convergence of quasi-Newton methods. *IMA Journal of Applied Mathematics*, 12(3): 223–245.

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

Chang, D.; Sun, S.; and Zhang, C. 2019. An accelerated linearly convergent stochastic L-BFGS algorithm. *IEEE Transactions on neural networks and learning systems*, 30(11): 3338–3346.

Conn, A. R.; Gould, N. I.; and Toint, P. L. 1991. Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Mathematical programming*, 50(1-3): 177–195.

Davidon, W. C. 1991. Variable metric method for minimization. *SIAM Journal on optimization*, 1(1): 1–17.

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Twenty-seventh Conference on Neural Information Processing Systems*.

Dennis, J. E.; and Moré, J. J. 1974. A characterization of superlinear convergence and its application to quasi-Newton methods. *Mathematics of computation*, 28(126): 549–560.

Dixon, L. 1972a. Quasi-Newton algorithms generate identical points. *Mathematical Programming*, 2: 383–387.

Dixon, L. 1972b. Quasi Newton techniques generate identical points II: the proofs of four new theorems. *Mathematical Programming*, 3: 345–358.

Fletcher, R. 1970. A new approach to variable metric algorithms. *The computer journal*, 13(3): 317–322.

Fletcher, R.; and Powell, M. J. 1963. A rapidly convergent descent method for minimization. *The computer journal*, 6(2): 163–168.

Gao, W.; and Goldfarb, D. 2018. Block BFGS methods. *SIAM Journal on Optimization*, 28(2): 1205–1231.

Gao, Z.; Koppel, A.; and Ribeiro, A. 2020. Incremental greedy BFGS: An incremental quasi-Newton method with explicit superlinear rate. In *Advanced Neural Information Processing System 12th OPT Workshop Optimization on Machine Learning*.

Gay, D. M. 1979. Some convergence properties of Broyden's method. *SIAM Journal on Numerical Analysis*, 16(4): 623–630.

Goldfarb, D. 1970. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109): 23–26.

Gower, R.; Goldfarb, D.; and Richtárik, P. 2016. Stochastic block BFGS: Squeezing more curvature out of data. In *International Conference on Machine Learning*, 1869–1878. PMLR.

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

Hastie, T.; Tibshirani, R.; Friedman, J. H.; and Friedman, J. H. 2009. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Jin, Q.; and Mokhtari, A. 2023. Non-asymptotic superlinear convergence of standard quasi-Newton methods. *Mathematical Programming*, 200(1): 425–473.

Johnson, R.; and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Twenty-sixth Conference on Neural Information Processing Systems*.

Lahoti, A.; Senapati, S.; Rajawat, K.; and Koppel, A. 2023. Sharpened Lazy Incremental Quasi-Newton Method. *arXiv preprint arXiv:2305.17283*.

Lin, D.; Ye, H.; and Zhang, Z. 2022. Explicit convergence rates of greedy and random quasi-Newton methods. *The Journal of Machine Learning Research*, 23(1): 7272–7311.

Liu, C.; Bi, S.; Luo, L.; and Lui, J. C. 2022. Partial-Quasi-Newton Methods: Efficient Algorithms for Minimax Optimization Problems with Unbalanced Dimensionality. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1031–1041.

Liu, C.; Chen, C.; and Luo, L. 2023. Symmetric Rank-k Methods. *arXiv preprint arXiv:2303.16188*.

Liu, C.; Chen, C.; Luo, L.; and Lui, J. C. 2023. Block Broyden's Methods for Solving Nonlinear Equations. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Liu, C.; and Luo, L. 2022. Quasi-Newton Methods for Saddle Point Problems. In *Thirty-fifth Conference on Neural Information Processing Systems*.

Lucchi, A.; McWilliams, B.; and Hofmann, T. 2015. A variance reduced stochastic Newton method. *arXiv preprint arXiv:1503.08316*.

Mokhtari, A.; Eisen, M.; and Ribeiro, A. 2018. IQN: An incremental quasi-Newton method with local superlinear convergence rate. *SIAM Journal on Optimization*, 28(2): 1670–1698.

Mokhtari, A.; and Ribeiro, A. 2014. RES: Regularized stochastic BFGS algorithm. *IEEE Transactions on Signal Processing*, 62(23): 6089–6104.

Mokhtari, A.; and Ribeiro, A. 2015. Global convergence of online limited memory BFGS. *The Journal of Machine Learning Research*, 16(1): 3151–3181.

Moritz, P.; Nishihara, R.; and Jordan, M. 2016. A linearly-convergent stochastic L-BFGS algorithm. In *Artificial Intelligence and Statistics*, 249–258. PMLR.

Murphy, K. P. 2012. *Machine learning: a probabilistic perspective*. MIT press.

Nesterov, Y. 2003. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Nocedal, J.; and Wright, S. J. 1999. *Numerical optimization*. Springer.

O'Leary, D. P.; and Yeremin, A. 1994. The linear algebra of block quasi-Newton algorithms. *Linear Algebra and its Applications*, 212: 153–168.

Powell, M. J. 1971. On the convergence of the variable metric algorithm. *IMA Journal of Applied Mathematics*, 7(1): 21–36.

Rodomanov, A.; and Kropotov, D. 2016. A superlinearly-convergent proximal Newton-type method for the optimization of finite sums. In *International Conference on Machine Learning*, 2597–2605. PMLR.

Rodomanov, A.; and Nesterov, Y. 2021a. Greedy quasi-Newton methods with explicit superlinear convergence. *SIAM Journal on Optimization*, 31(1): 785–811.

Rodomanov, A.; and Nesterov, Y. 2021b. New results on superlinear convergence of classical quasi-Newton methods. *Journal of optimization theory and applications*, 188: 744–769.

Rodomanov, A.; and Nesterov, Y. 2021c. Rates of superlinear convergence for classical quasi-Newton methods. *Mathematical Programming*, 1–32.

Schmidt, M.; Le Roux, N.; and Bach, F. 2017. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162: 83–112.

Schnabel, R. B. 1983. Quasi-Newton methods using multiple secant equations. *Computer Science Technical Reports*, 244(41): 06.

Shanno, D. F. 1970. Conditioning of quasi-Newton methods for function minimization. *Mathematics of computation*, 24(111): 647–656.

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

Woodworth, B. E.; and Srebro, N. 2016. Tight complexity bounds for optimizing composite objectives. In *Twenty-ninth Conference on Neural Information Processing Systems*.

Yang, M.; Milzarek, A.; Wen, Z.; and Zhang, T. 2021. A stochastic extra-step quasi-Newton method for nonsmooth nonconvex optimization. *Mathematical Programming*, 1–47.

Ye, H.; Lin, D.; Zhang, Z.; and Chang, X. 2021. Explicit superlinear convergence rates of the SR1 algorithm. *arXiv preprint arXiv:2105.07162*.

Zhang, L.; Mahdavi, M.; and Jin, R. 2013. Linear convergence with condition number independent access of full gradients. In *Twenty-sixth Conference on Neural Information Processing Systems*.