# Greedy and Random Quasi-Newton Methods with Faster Explicit Superlinear Convergence

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

In this paper we follow Rodomanov and Nesterov [18]'s work to study quasi-Newton methods. We focus on the common SR1 and BFGS quasi-Newton methods to establish better explicit (local) superlinear convergence. First, based on *greedy quasi-Newton update* which greedily selects the direction so as to maximize a certain measure of progress, we improve the convergence rate to a condition-number-free superlinear convergence rate. Second, based on *random quasi-Newton update* that selects the direction randomly from a spherically symmetric distribution, we show the same superlinear convergence rate established as above. Our analysis is closely related to the approximation of a given Hessian matrix, unconstrained quadratic objective, as well as the general strongly convex, smooth and strongly self-concordant functions.

## 1 Introduction

**Motivation.** We investigate the superlinear convergence of famous quasi-Newton methods, that replace the exact Hessian applied in classical Newton methods to certain approximation. And the approximation is updated in iterations based on some special formulas from previous variation. There exist various quasi-Newton algorithms employing different Hessian approximations. The three most popular versions are the *Davidon-Fletcher-Powell (DFP) method* [7, 10], the *Broyden-Fletcher-Goldfarb-Shanno (BFGS) method* [2, 3, 9, 11, 20], and the *Symmetric Rank 1 (SR1) method* [1, 7], all of which belong to the Broyden family [1] of quasi-Newton algorithms. The most attractive feature of quasi-Newton methods is their superlinear convergence, which can trace back to 1970s [4, 8, 16]. However, the superlinear convergence rates provided in previous work are only asymptotic [5, 13, 14, 21, 23], because their results only show that the ratio of successive residuals in these methods tends to zero when the running iterations turn to infinity, while still lacking of a specific superlinear convergence rate.

Recently, Rodomanov and Nesterov [18] gave the first explicit local superlinear convergence with their proposed new quasi-Newton methods. They introduced greedy quasi-Newton update by greedily selecting from basis vectors to maximize a certain measure of progress. They also established an explicit nonasymptotic bound on their rate of local superlinear convergence correspondingly. However, their proofs are mainly applied to DFP methods because they reduced all possible Broyden family to the DFP update based on the monotonicity property (see Lemma 2.2). In addition, the SR1 and BFGS update are more popular and faster than the DFP update in practice, which also has been verified in their experiments. Moreover, they also discovered that the randomized methods do not loose superlinear convergence but they do not provide theoretical guarantees.

There are other analyses of local superlinear convergence along the same line of research. Rodomanov and Nesterov [19] also analyzed the well-known DFP and BFGS methods, adopting standard Hessian update direction through the previous variation. They demonstrated faster initial convergence

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

| | Strongly Self-concordant Objective | $k_0$ |
|---|---|---|
| DFP, BFGS, SR1 [18] | $\left(1-\frac{1}{n\kappa}\right)^{k(k-1)/2}\left(\frac{1}{2}\right)^{k}\left(1-\frac{1}{2\kappa}\right)^{k_0}$ | $O\left(n\kappa\ln(n\kappa)\right)$ |
| BFGS (ours, Thm 4.2) | $\left(1-\frac{1}{n}\right)^{k(k-1)/2}\left(\frac{1}{2}\right)^{k}\left(1-\frac{1}{2\kappa}\right)^{k_0}$ | $O\left(\max\{n,\kappa\}\ln(n\kappa)\right)$ |
| SR1 (ours, Thm 4.3) | $\left(1-\frac{1}{n}\right)^{k(k-1)/2}\left(\frac{1}{2}\right)^{k}\left(1-\frac{1}{2\kappa}\right)^{k_0}$ | $O\left(\max\{n,\kappa\}\ln(n\kappa)\right)$ |

Table 1: Comparison of existing specific superlienar convergence rate of random or greedy quasi-Newton methods with our results in the view of $\lambda_f(\cdot)$ (see the definition in Eq. (14)), where $n$ is the dimension of parameters, $\kappa$ is the condition number of the objective function, and $k_0, k$ are the iteration numbers. Rodomanov and Nesterov [18]'s results are suitable for greedy quasi-Newton methods using Broyden family update, which includes DFP, BFGS, SR1 methods.

rates, while slower final rates compared to Rodomanov and Nesterov [18]'s results. Additionally, Rodomanov and Nesterov [17] improved Rodomanov and Nesterov [19]'s results by reducing the dependence of the condition number $\kappa$ to $\ln\kappa$, though having similar worse long history behavior.

In this work, we close the gap in the experiments of [18] by proving faster convergence rates for greedy and random quasi-Newton algorithms. We present our contribution as follows:

- First, in the setting of approximating an exact Hessian, we use two update methods: 1) *greedy update* modified from Rodomanov and Nesterov [18]; 2) *random update* that selects from any spherically symmetric distribution. We show that both the methods share a faster condition-number-free convergence. Particularly, we obtain the superlinear convergence rate $O\left(1-\frac{k}{n}\right)$ for SR1 update, and the linear convergence rate $O((1-\frac{1}{n})^k)$ for BFGS update, where $k$ is the current iteration, and $n$ is the dimension of parameters. Both findings improve the original convergence rate $O((1-\frac{1}{n\kappa})^k)$ [18].
- Second, we extent our analysis to practical scheme, showing (local) superlinear convergence under the SR1 update and BFGS update, when applied to unconstrained quadratic objective or strongly self-concordant functions. We list our results in Table 1 with same formulation of [18]. Note that in general, the convergence goes through two periods. The first period lasts for $k_0$ iterations, and only has a linear convergence rate $O((1-\frac{1}{2\kappa})^{k_0})$. The second period has a superlinear convergence rate $O((1-\frac{1}{n})^{k(k-1)/2})$. Our revised bound takes less first period iterations $k_0$ as well as a faster (condition-number-free) superlinear convergence rate in the second period compared to [18].

**Notation.** We denote vectors by lowercase bold letters (e.g., $\boldsymbol{u}, \boldsymbol{x}$), and matrices by capital bold letters (e.g., $\boldsymbol{W} = [w_{ij}]$). We use $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ for the $n$-dimensional coordinate directions. Let $\lambda_1(\boldsymbol{A}) \geq \cdots \geq \lambda_n(\boldsymbol{A})$ be the eigenvalues of a real symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. Moreover, $\|\cdot\|$ denotes the $\ell_2$-norm (standard Euclidean norm). For a given positive definite matrix $\boldsymbol{A}$ (i.e., $\boldsymbol{A} \succ 0$), we induce the following pair of conjugate Euclidean norms: $\|\boldsymbol{x}\|_{\boldsymbol{A}} \triangleq \sqrt{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x}}$, $\|\boldsymbol{x}\|^*_{\boldsymbol{A}} \triangleq \sqrt{\boldsymbol{x}^\top \boldsymbol{A}^{-1}\boldsymbol{x}}$. When $\boldsymbol{A} = \nabla^2 f(\boldsymbol{x}) \succ 0$, we prefer to use notation $\|\cdot\|_{\boldsymbol{x}}$ and $\|\cdot\|^*_{\boldsymbol{x}}$, provided that there is no ambiguity with the reference function $f$. We say the condition number of a positive definite matrix $\boldsymbol{A}$ as $\kappa(\boldsymbol{A}) = \lambda_1(\boldsymbol{A})/\lambda_n(\boldsymbol{A})$, or simplified as $\kappa$ if there is no ambiguity. Finally, we recall the definition of the rate of convergence used in this paper.

**Definition 1.1 (Linear / Superlinear convergence)** *Suppose a sequence $\{\boldsymbol{x}_n\}$ converges to $\boldsymbol{x}_\infty$ that satisfies*

$$\lim_{n\to\infty} \frac{\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_\infty\|}{\|\boldsymbol{x}_n - \boldsymbol{x}_\infty\|} = q \in [0, 1).$$

*We say the sequence $\{\boldsymbol{x}_n\}$ converges superlinearly if $q = 0$, linearly if $q \in (0, 1)$.*

**Organization.** In Section 2, we discuss quasi-Newton updating rules for approximating positive definite Hessian matrix using the SR1 update (Subsection 2.1) and the BFGS update (Subsection 2.2). In Section 3, we analyze previous methods applied to the problem of minimizing a quadratic function. In Section 4, we show that similar results also hold in a more general setting of minimizing the strongly self-concordant function. We give some empirical results in Section 5. Finally, in Section 6, we present our conclusion.

## 2 Quasi-Newton updates

Before starting our theoretical results, we briefly review a class of quasi-Newton updating rules for approximating a positive definite matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. We follow the definition in [18], adopting the following family of updates which describes the Broyden family [15, Section 6.3] of quasi-Newton updates, parameterized by a scalar $\tau \in \mathbb{R}$.

**Definition 2.1** *Let $\boldsymbol{A} \preceq \boldsymbol{G}$ (that is, $\boldsymbol{G} - \boldsymbol{A}$ is positive semi-definite). For any $\boldsymbol{u} \in \mathbb{R}^n$, if $\boldsymbol{G}\boldsymbol{u} = \boldsymbol{A}\boldsymbol{u}$, define $\mathrm{Broyd}_\tau(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}) \triangleq \boldsymbol{G}$. Otherwise, we define*

$$\begin{aligned}\mathrm{Broyd}_\tau(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}) \triangleq \tau &\left[\boldsymbol{G} - \frac{\boldsymbol{A}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{G} + \boldsymbol{G}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{A}}{\boldsymbol{u}^\top\boldsymbol{A}\boldsymbol{u}} + \left(\frac{\boldsymbol{u}^\top\boldsymbol{G}\boldsymbol{u}}{\boldsymbol{u}^\top\boldsymbol{A}\boldsymbol{u}} + 1\right)\frac{\boldsymbol{A}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{A}}{\boldsymbol{u}^\top\boldsymbol{A}\boldsymbol{u}}\right] \\ &+ (1-\tau)\left[\boldsymbol{G} - \frac{(\boldsymbol{G}-\boldsymbol{A})\boldsymbol{u}\boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})}{\boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})\boldsymbol{u}}\right].\end{aligned} \tag{1}$$

For several choices of $\tau$, we can recover several well-known quasi-Newton methods.

For $\tau = 0$, Eq. (1) corresponds to the well-known SR1 update:

$$\mathrm{SR1}(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}) \triangleq \boldsymbol{G} - \frac{(\boldsymbol{G}-\boldsymbol{A})\boldsymbol{u}\boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})}{\boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})\boldsymbol{u}}. \tag{2}$$

For $\tau_{BFGS} \triangleq \frac{\boldsymbol{u}^\top\boldsymbol{A}\boldsymbol{u}}{\boldsymbol{u}^\top\boldsymbol{G}\boldsymbol{u}} \in [0, 1]$, we recover the famous BFGS update:

$$\mathrm{BFGS}(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}) \triangleq \boldsymbol{G} - \frac{\boldsymbol{G}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{G}}{\boldsymbol{u}^\top\boldsymbol{G}\boldsymbol{u}} + \frac{\boldsymbol{A}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{A}}{\boldsymbol{u}^\top\boldsymbol{A}\boldsymbol{u}}. \tag{3}$$

For $\tau = 1$, it corresponds to the well-known DFP update:

$$\mathrm{DFP}(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}) \triangleq \boldsymbol{G} - \frac{\boldsymbol{A}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{G} + \boldsymbol{G}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{A}}{\boldsymbol{u}^\top\boldsymbol{A}\boldsymbol{u}} + \left(\frac{\boldsymbol{u}^\top\boldsymbol{G}\boldsymbol{u}}{\boldsymbol{u}^\top\boldsymbol{A}\boldsymbol{u}} + 1\right)\frac{\boldsymbol{A}\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{A}}{\boldsymbol{u}^\top\boldsymbol{A}\boldsymbol{u}}. \tag{4}$$

Additionally, the Broyden family has benign monotonicity below, showing the relationship between these quasi-Newton methods.

**Lemma 2.2 (Rodomanov and Nesterov [18] Lemmas 2.1 and 2.2)** *If $\boldsymbol{A} \preceq \boldsymbol{G} \preceq \eta\boldsymbol{A}$ for some $\eta \geq 1$, then we have for any $\boldsymbol{u} \in \mathbb{R}^n$, and $\tau_1, \tau_2 \in \mathbb{R}$ with $\tau_1 \leq \tau_2$ that*

$$\mathrm{Broyd}_{\tau_1}(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}) \preceq \mathrm{Broyd}_{\tau_2}(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}).$$

*And for any $\tau \in [0, 1]$, we have $\boldsymbol{A} \preceq \mathrm{Broyd}_\tau(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}) \preceq \eta\boldsymbol{A}$.*

Hence, from Lemma 2.2, if $\boldsymbol{A} \preceq \boldsymbol{G} \preceq \eta\boldsymbol{A}$ for some $\eta \geq 1$, it follows that

$$\boldsymbol{A} \preceq SR1(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}) \preceq BFGS(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}) \preceq DFP(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}) \preceq \eta\boldsymbol{A}. \tag{5}$$

Intuitively, the approximation produced by SR1, is better than the one produced by BFGS. And both of them is better than the one produced by DFP. Therefore, there may also have different superlinear convergence rates behind them. Since Rodomanov and Nesterov [18] deduced all methods to the slowest DFP update, we demonstrate the other two providing better superlinear convergence rates.

### 2.1 Superlinear convergence for SR1 update

Here we consider the classical quasi-Newton updating rules SR1 update for approximating a fixed positive definite matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and give the proof in Appendix B.

Let us now justify the efficiency of update Eq. (2) in ensuring convergence $\boldsymbol{G} \to \boldsymbol{A}$. For this, we introduce the following measure of progress:

$$\tau_{\boldsymbol{A}}(\boldsymbol{G}) \triangleq \mathrm{tr}(\boldsymbol{G} - \boldsymbol{A}). \tag{6}$$

Previous work [18] adopts a different measure as follows

$$\sigma_{\boldsymbol{A}}(\boldsymbol{G}) \triangleq \langle \boldsymbol{A}^{-1}, \boldsymbol{G}\rangle - n = \mathrm{tr}\left(\boldsymbol{A}^{-1}(\boldsymbol{G} - \boldsymbol{A})\right), \tag{7}$$

**Algorithm 1** SR1 update

Initialization: set $\boldsymbol{G}_0 \succ \boldsymbol{A}$.
**for** $n-1 \geq k \geq 0$ **do**
  Choose $\boldsymbol{u}_k$ from
  1) *greedy method* (Eq. 9), or
  2) *random method* (Eq. 10).
  Set $\boldsymbol{G}_{k+1} = \mathrm{SR1}(\boldsymbol{G}_k, \boldsymbol{A}, \boldsymbol{u}_k)$.
**end for**

**Algorithm 2** BFGS update

Initialization: set $\boldsymbol{G}_0 \succ \boldsymbol{A}, \boldsymbol{L}_0 = \boldsymbol{G}_0^{-1/2}$.
**for** $k \geq 0$ **do**
  Compute $\boldsymbol{u}_k = \boldsymbol{L}_k \tilde{\boldsymbol{u}}_k$ with $\tilde{\boldsymbol{u}}_k$ from
  1) *greedy method* (Eq. 12), or
  2) *random method* (Eq. 10).
  Set $\boldsymbol{G}_{k+1} = \mathrm{BFGS}(\boldsymbol{G}_k, \boldsymbol{A}, \boldsymbol{u}_k)$.
  Compute $\boldsymbol{L}_{k+1}$ following Lemma C.1.
**end for**

which might be improper for analyzing the SR1 update. Thus we employ a more concise measure $\tau_{\boldsymbol{A}}(\cdot)$. However, we would reuse $\sigma_{\boldsymbol{A}}(\cdot)$ in the proof of BFGS update in Subsection 2.2.

According to $\tau_{\boldsymbol{A}}(\boldsymbol{G})$, one iteration update leads to

$$\tau_{\boldsymbol{A}}(\boldsymbol{G}_+) = \tau_{\boldsymbol{A}}(\boldsymbol{G}) - \frac{\boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})^2\boldsymbol{u}}{\boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})\boldsymbol{u}}, \quad \boldsymbol{G}_+ = \mathrm{SR1}(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}). \tag{8}$$

Note that we always have $\boldsymbol{G}_+ \succeq \boldsymbol{A}$ if $\boldsymbol{G} \succeq \boldsymbol{A}$ from Lemma 2.2. Moreover, the choice of the updating direction $\boldsymbol{u}$ directly influences the decreasing in the measure $\tau_{\boldsymbol{A}}(\cdot)$. In the following, we show two efficient ways for selecting $\boldsymbol{u}$ introduced in Algorithm 1.

First, we use *greedy method* as given in [18], that greedily selects $\boldsymbol{u}$ from the basis vectors to obtain the largest decreasing of $\tau_{\boldsymbol{A}}(\boldsymbol{G}_+) - \tau_{\boldsymbol{A}}(\boldsymbol{G})$: $\bar{\boldsymbol{u}}_{\boldsymbol{A}}(\boldsymbol{G}) \triangleq \arg\max_{\boldsymbol{u}\in\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_n\}} \frac{\boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})^2\boldsymbol{u}}{\boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})\boldsymbol{u}}$. However, we may encounter numerical overflow due to dividing zero if $\boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})\boldsymbol{u}$ is nearly zero. Notice that $\boldsymbol{G} \succeq \boldsymbol{A}$, we have $\sqrt{(\boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})^2\boldsymbol{u})\,(\boldsymbol{u}^\top\boldsymbol{u})} \geq \boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})\boldsymbol{u} \geq 0,\ \forall \boldsymbol{u} \in \mathbb{R}^n$ from the Cauchy–Schwarz inequality. Thus we employ a safer adjustment below:

$$\bar{\boldsymbol{u}}_{\boldsymbol{A}}(\boldsymbol{G}) \triangleq \operatorname*{arg\,max}_{\boldsymbol{u}\in\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_n\}} \frac{\boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})\boldsymbol{u}}{\boldsymbol{u}^\top\boldsymbol{u}} = \operatorname*{arg\,max}_{\boldsymbol{u}\in\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_n\}} \boldsymbol{u}^\top(\boldsymbol{G}-\boldsymbol{A})\boldsymbol{u}. \tag{9}$$

Moreover, we only need to obtain the diagonal elements of $\boldsymbol{A}$ (the current Hessian in practice), thus the complexity is $O(n^2)$[1], which is acceptable and same as the classical quasi-Newton methods.

Second, we use *random method*, i.e., choosing $\boldsymbol{u}$ from a spherically symmetric distribution, e.g.,

$$\boldsymbol{u} \sim \mathcal{N}(0, I_n) \text{ or } \boldsymbol{u} \sim \mathrm{Unif}(\mathcal{S}^{n-1}), \tag{10}$$

leading to a stochastic algorithm with running complexity $O(n^2)$ similarly.

Rodomanov and Nesterov [18, Theorem 3.5] already showed that $\boldsymbol{G}_n = \boldsymbol{A}$ if the $\boldsymbol{u}_k$ for $k \in \{0, \ldots, n-1\}$ are linearly independent. Thus in the following, we only consider $k \leq n$. Now let us estimate the decrease in the measure $\tau_{\boldsymbol{A}}(\cdot)$ based on the direction $\boldsymbol{u}$ chosen from previous strategies.

**Theorem 2.3** *Under the SR1 update in Algorithm 1, we can obtain that*

1) *for greedy method with $\boldsymbol{u}_k$ chosen from the basis vectors,*

$$0 \preceq \boldsymbol{G}_k - \boldsymbol{A} \preceq \left(1 - \frac{k}{n}\right)\tau_{\boldsymbol{A}}(\boldsymbol{G}_0)\boldsymbol{I}_n,\ 0 \leq k \leq n;$$

2) *for random method with $\boldsymbol{u}_k$ chosen from a spherically symmetric distribution,*

$$0 \preceq \mathbb{E}_{\boldsymbol{u}_{k-1},\ldots,\boldsymbol{u}_0}\left[\boldsymbol{G}_k - \boldsymbol{A}\right] \preceq \left(1 - \frac{k}{n}\right)\tau_{\boldsymbol{A}}(\boldsymbol{G}_0)\boldsymbol{I}_n,\ 0 \leq k \leq n.$$

*Hence, $\boldsymbol{G}_k$ (or $\mathbb{E}_{\boldsymbol{u}_{k-1},\ldots,\boldsymbol{u}_0}\boldsymbol{G}_k$) converges superlinearly until it becomes the exact Hessian matrix $\boldsymbol{A}$.*

**Remark 2.4** *In fact, for random method, when applied to $\sigma_{\boldsymbol{A}}(\cdot)$, we could not handle the expectation of the division of random variables with known expectations, because the random variables themselves are non-independent as Rodomanov and Nesterov [18] mentioned. We take $\tau_{\boldsymbol{A}}(\cdot)$ instead, which overcomes this difficulty due to the same spectral decomposition of $(\boldsymbol{G}-\boldsymbol{A})^2$ and $\boldsymbol{G}-\boldsymbol{A}$.*

[1] Note that we can use Hessian-vector product to obtain $\boldsymbol{A}\boldsymbol{u}$ in practice.

**Algorithm 3** Quasi-Newton methods for quadratic minimization

Initialization: set $\boldsymbol{G}_0 = L\boldsymbol{I}_n$ (or any $\boldsymbol{G}_0 \succ \boldsymbol{A}$), $\boldsymbol{H}_0 = \boldsymbol{G}_0^{-1}$.
**for** $k \geq 0$ **do**
  Update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{H}_k \nabla f(\boldsymbol{x}_k)$.
  Choose one of the following update rule:
  (i) SR1: Choose $\boldsymbol{u}_k$ following Algorithm 1.
  Set $\boldsymbol{G}_{k+1} = \mathrm{SR1}(\boldsymbol{G}_k, \boldsymbol{A}, \boldsymbol{u}_k)$, $\boldsymbol{H}_{k+1} = \mathrm{InvSR1}(\boldsymbol{H}_k, \boldsymbol{A}, \boldsymbol{u}_k)$.
  (ii) BFGS: Choose $\boldsymbol{u}_k$ following Algorithm 2.
  Set $\boldsymbol{G}_{k+1} = \mathrm{BFGS}(\boldsymbol{G}_k, \boldsymbol{A}, \boldsymbol{u}_k)$, $\boldsymbol{H}_{k+1} = \mathrm{InvBFGS}(\boldsymbol{H}_k, \boldsymbol{A}, \boldsymbol{u}_k)$.
**end for**

### 2.2 Linear convergence for BFGS update

We now consider another classical quasi-Newton BFGS update in the same scheme and leave the proof in Appendix B. Using the measure $\sigma_{\boldsymbol{A}}(\cdot)$, we obtain that

$$\sigma_{\boldsymbol{A}}(\boldsymbol{G}_+) = \sigma_{\boldsymbol{A}}(\boldsymbol{G}) - \frac{\boldsymbol{u}^\top \boldsymbol{G}\boldsymbol{A}^{-1}\boldsymbol{G}\boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{G}\boldsymbol{u}} + 1,\ \boldsymbol{G}_+ = \mathrm{BFGS}(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{u}). \tag{11}$$

If we directly apply *greedy method* or *random method* from the previous content, we could only obtain the same linear convergence rate $\sigma_{\boldsymbol{A}}(\boldsymbol{G}_+) \leq \left(1 - \frac{1}{n\kappa}\right)\sigma_{\boldsymbol{A}}(\boldsymbol{G})$ as Rodomanov and Nesterov [18, Theorem 2.5]. However, if we take advantage of the current $\boldsymbol{G}$, and choose a scaled direction such that $\boldsymbol{u} = \boldsymbol{G}^{-1/2}\tilde{\boldsymbol{u}}$, we could simplify the formulation and obtain a faster condition-number-free linear convergence rate. Then the greedy update after replacing $\boldsymbol{u}$ to $\boldsymbol{G}^{-1/2}\tilde{\boldsymbol{u}}$ is as follows:

$$\tilde{\boldsymbol{u}}_{\boldsymbol{A}}(\boldsymbol{G}) = \operatorname*{arg\,max}_{\tilde{\boldsymbol{u}} \in \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}} \frac{\tilde{\boldsymbol{u}}^\top \boldsymbol{G}^{1/2}\boldsymbol{A}^{-1}\boldsymbol{G}^{1/2}\tilde{\boldsymbol{u}}}{\tilde{\boldsymbol{u}}^\top \tilde{\boldsymbol{u}}} = \operatorname*{arg\,max}_{\tilde{\boldsymbol{u}} \in \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}} \tilde{\boldsymbol{u}}^\top \boldsymbol{G}^{1/2}\boldsymbol{A}^{-1}\boldsymbol{G}^{1/2}\tilde{\boldsymbol{u}}. \tag{12}$$

Now we give the linear convergence rate of the BFGS update with our direction choosing methods.

**Theorem 2.5** *Under the BFGS update in Algorithm 2, we can obtain that for all $k \geq 0$,*

1) *for greedy method with $\tilde{\boldsymbol{u}}_k$ chosen from the basis vectors,*

$$0 \preceq \boldsymbol{G}_k - \boldsymbol{A} \preceq \left(1 - \frac{1}{n}\right)^k \sigma_A(\boldsymbol{G}_0)\boldsymbol{A};$$

2) *for random method with $\tilde{\boldsymbol{u}}_k$ chosen from a spherically symmetric distribution,*

$$0 \preceq \mathbb{E}_{\boldsymbol{u}_{k-1}, \ldots, \boldsymbol{u}_0}\left[\boldsymbol{G}_k - \boldsymbol{A}\right] \preceq \left(1 - \frac{1}{n}\right)^k \sigma_{\boldsymbol{A}}(\boldsymbol{G}_0)\boldsymbol{A}.$$

*Therefore, $\boldsymbol{G}_k$ (or $\mathbb{E}_{\boldsymbol{u}_{k-1}, \ldots, \boldsymbol{u}_0}\boldsymbol{G}_k$) converges to the exact Hessian matrix $\boldsymbol{A}$ linearly.*

**Remark 2.6** *Note that the complexity in Eq. (12) is $O(n^3)$ because we have multiply-add operations with $\boldsymbol{A}^{-1}$. Hence we do not apply this greedy strategy in practice, but view it as a theoretical result similar as the random strategy. Moreover, for the random method, we also show the efficiency of our scaled direction compared to the original direction in our experiments.*

Finally, we are able to use an efficient way (with complexity $O(n^2)$) for updating $\boldsymbol{G}^{-1/2}$ shown in Lemma C.1. The main idea is employing rank-one update and one row appending update of QR decomposition [12, Section 12.5.1 and 12.5.3] with inverse BFGS update.

## 3 Unconstrained quadratic minimization

Now let us demonstrate how we can apply the SR1 and BFGS updates described in the previous section, for minimizing the quadratic function:

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}, \tag{13}$$

where $\boldsymbol{A}$ is a positive definite linear operator, and there exist $\mu, L > 0$, s.t., $\mu \boldsymbol{I}_n \preceq \boldsymbol{A} \preceq L\boldsymbol{I}_n$. The algorithm is shown in Algorithm 3. As classical quasi-Newton methods do, we need to use quasi-Newton step for updating the parameters as well as approximating the true Hessian matrix. Moreover, we use inverse update rules (InvSR1 and InvBFGS) to replace $\boldsymbol{G}_k^{-1}$ as $\boldsymbol{H}_k$ [2]. To estimate the convergence rate of Scheme (13), let us look at the norm of the gradient of $f$, measured with

$$\lambda_f(\boldsymbol{x}) \triangleq \sqrt{\nabla f(\boldsymbol{x})^\top \boldsymbol{A}^{-1} \nabla f(\boldsymbol{x})}, \boldsymbol{x} \in \mathbb{R}^n. \tag{14}$$

Note that this measure of optimality is directly related to the functional residual. Indeed, let $\boldsymbol{x}^* = \boldsymbol{A}^{-1}\boldsymbol{b}$ be the minimizer of Eq. (13). Then we obtain

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^*) = \frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{x}^*\right)^\top \boldsymbol{A}\left(\boldsymbol{x} - \boldsymbol{x}^*\right) = \frac{1}{2}\left(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\right)^\top \boldsymbol{A}^{-1}\left(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\right) = \frac{1}{2}\lambda_f(\boldsymbol{x})^2.$$

The following lemma shows how $\lambda_f(\cdot)$ changes after one iteration of process in Algorithm 3.

**Lemma 3.1 (Rodomanov and Nesterov [18] Lemma 3.2)** *Let $k \geq 0$, and let $\eta_k \geq 1$ be such that $\boldsymbol{A} \preceq \boldsymbol{G}_k \preceq \eta_k \boldsymbol{A}$. Then we have that $\lambda_f(\boldsymbol{x}_{k+1}) \leq \left(1 - \frac{1}{\eta_k}\right)\lambda_f(\boldsymbol{x}_k) \leq (\eta_k - 1)\lambda_f(\boldsymbol{x}_k)$.*

Thus, to estimate how fast $\lambda_f(\boldsymbol{x}_k)$ converges to zero, we need to upper bound $\eta_k$, which was already done in Subsections 2.1 and 2.2. Therefore, we can guarantee a superlinear convergence of $\lambda_f(\boldsymbol{x}_k)$ using the SR1 or the BFGS update. The proof of Theorem 3.2 can be found in Appendix B.

**Theorem 3.2** *Under Algorithm 3, if we choose SR1 update, then we have*

$$\mathbb{E}\lambda_f(\boldsymbol{x}_{k+1}) \leq \left(1 - \frac{k}{n}\right)\frac{\operatorname{tr}(\boldsymbol{G}_0 - \boldsymbol{A})}{\mu}\mathbb{E}\lambda_f(\boldsymbol{x}_k),\ 0 \leq k \leq n.$$

*If we choose BFGS update, then we have*

$$\mathbb{E}\lambda_f(\boldsymbol{x}_{k+1}) \leq \left(1 - \frac{1}{n}\right)^k \operatorname{tr}\left(\left(\boldsymbol{G}_0 - \boldsymbol{A}\right)\boldsymbol{A}^{-1}\right)\mathbb{E}\lambda_f(\boldsymbol{x}_k),\ k \geq 0.$$

*The expectation considers all the randomness of the directions $\boldsymbol{u}_k$'s during iterations, and when applied to the greedy method, we can view it with no randomness for the same notation.*

Therefore, we could see both SR1 and BFGS updates share superlinear convergence for $f(\boldsymbol{x}_k) \to f(\boldsymbol{x}^*)$. In particular, for SR1 update, our bound indeed recovers the classical result in [15, Theorem 6.1], showing that the update stops after finite steps because $\boldsymbol{G}_n = \boldsymbol{A}$ and $\lambda_f(\boldsymbol{x}_{n+1}) = 0$. Moreover, we also give an exact convergence rate during the whole training process. And the main decreasing term $\left(1 - \frac{k}{n}\right)$ for SR1 update, as well as $(1 - \frac{1}{n})^k$ for BFGS update in $k$-th iteration is independent of the condition number $\kappa$ of $\boldsymbol{A}$, which improves the bound $(1 - \frac{1}{n\kappa})^k$ in [18, Theorem 3.4].

# 4 Minimization of general functions

Now we consider a general problem of unconstrained minimization: $\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$, where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice differentiable function with positive definite Hessian. Our goal is to extend the results in the previous sections, assuming that the methods can start from a sufficiently good initial point $\boldsymbol{x}_0$. We use the same assumption *strongly self-concordant* followed by Rodomanov and Nesterov [18],

**Definition 4.1 (Strongly self-concordant)** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is strongly self-concordant if the Hessians of $f$ are close to each other in the sense that there exists a constant $M \geq 0$ s.t.*

$$\nabla^2 f(\boldsymbol{y}) - \nabla^2 f(\boldsymbol{x}) \preceq M\|\boldsymbol{y} - \boldsymbol{x}\|_{\boldsymbol{z}} \nabla^2 f(\boldsymbol{w}),\ \forall\, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{w} \in \mathbb{R}^n.$$

Rodomanov and Nesterov [18] have already mentioned several properties and examples of strongly self-concordant function, such as a strongly convex function with Lipschitz continuous Hessian. Let

[2] $\mathrm{InvSR1}(\boldsymbol{H}, \boldsymbol{A}, \boldsymbol{u}), \mathrm{InvBFGS}(\boldsymbol{H}, \boldsymbol{A}, \boldsymbol{u})$ are shown in Appendix A or see [15].

**Algorithm 4** Quasi-Newton Methods for general strongly self-concordant objective

Initialization: set $\boldsymbol{G}_0 = L\boldsymbol{I}_n$, $\boldsymbol{H}_0 = \boldsymbol{I}_n/L$.
**for** $k \geq 0$ **do**
  Update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{H}_k \nabla f(\boldsymbol{x}_k)$.
  Compute $r_k = \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_{\boldsymbol{x}_k}$ and set $\tilde{\boldsymbol{G}}_k = (1 + Mr_k)\,\boldsymbol{G}_k$, $\tilde{\boldsymbol{H}}_k = \boldsymbol{H}_k/\,(1 + Mr_k)$.
  Choose one of the following update rules:
  (i) SR1 update: choose $\boldsymbol{u}_k$ as Algorithm 1, applied with $\boldsymbol{A} = \nabla^2 f(\boldsymbol{x}_{k+1}), \boldsymbol{G} = \tilde{\boldsymbol{G}}_k$.
  Compute $\boldsymbol{G}_{k+1} = \mathrm{SR1}(\tilde{\boldsymbol{G}}_k, \nabla^2 f(\boldsymbol{x}_{k+1}), \boldsymbol{u}_k)$, $\boldsymbol{H}_{k+1} = \mathrm{InvSR1}(\tilde{\boldsymbol{H}}_k, \nabla^2 f(\boldsymbol{x}_{k+1}), \boldsymbol{u}_k)$.
  (ii) BFGS update: choose $\boldsymbol{u}_k$ as Algorithm 2, applied with $\boldsymbol{A} = \nabla^2 f(\boldsymbol{x}_{k+1}), \boldsymbol{G} = \tilde{\boldsymbol{G}}_k$.
  Compute $\boldsymbol{G}_{k+1} = \mathrm{BFGS}(\tilde{\boldsymbol{G}}_k, \nabla^2 f(\boldsymbol{x}_{k+1}), \boldsymbol{u}_k)$, $\boldsymbol{H}_{k+1} = \mathrm{InvBFGS}(\tilde{\boldsymbol{H}}_k, \nabla^2 f(\boldsymbol{x}_{k+1}), \boldsymbol{u}_k)$.
**end for**

us now make one more common assumption about the function $f$ as Rodomanov and Nesterov [18] did, that the function $f$ is $\mu$-strongly convex and $L$-smooth, i.e., there exist $L, \mu > 0$ such that

$$\mu \boldsymbol{I}_n \preceq \nabla^2 f(\boldsymbol{x}) \preceq L\boldsymbol{I}_n, \forall\, \boldsymbol{x} \in \mathbb{R}^n. \tag{15}$$

Unlike quadratic minimization, the true Hessian in each step varies. In order to make $\boldsymbol{G}_k \succeq \nabla^2 f(\boldsymbol{x}_k)$ always hold, we need to adjust $\boldsymbol{G}_k$ before doing quasi-Newton update. Instructed from [18], we also use *correction strategy*, which enlarges the approximation $\boldsymbol{G}_k$ properly in the scheme of quasi-Newton methods in Algorithm 4. For simplicity, we assume that the constants $M$ and $L$ are available.

For BFGS update, we can analyze how the Hessian approximation measure $\sigma_{\boldsymbol{x}}(\boldsymbol{G}) \triangleq \sigma_{\nabla^2 f(\boldsymbol{x})}(\boldsymbol{G})$ changes after one iteration following [18]. The proof can be found in Appendix B.

**Theorem 4.2** *Suppose that in Algorithm 4 BFGS update is used, and that the initial point $\boldsymbol{x}_0$ is sufficiently close to the solution:*

$$M\lambda_f(\boldsymbol{x}_0) \leq \frac{\ln 2}{4\kappa(2n+1)}. \tag{16}$$

*Then for all $k \geq 0$, we have*

$$0 \preceq \boldsymbol{G}_k - \nabla^2 f(\boldsymbol{x}_k) \preceq 2n\kappa \left(1 - \frac{1}{n}\right)^k \nabla^2 f(\boldsymbol{x}_k), \tag{17}$$

*and*

$$\lambda_f(\boldsymbol{x}_{k+1}) \leq 2n\kappa \left(1 - \frac{1}{n}\right)^k \lambda_f(\boldsymbol{x}_k). \tag{18}$$

As for SR1 update, the main difficulty compared to [18] is that we utilize a different measure $\tau_{\boldsymbol{A}}(\cdot)$ instead of $\sigma_{\boldsymbol{A}}(\cdot)$, resulting in a refined proof version below, and we leave the proofs in Appendix B.

**Theorem 4.3** *From Algorithm 4 with SR1 update, suppose that the initial point $\boldsymbol{x}_0$ is sufficiently close to the solution:*

$$M\lambda_f(\boldsymbol{x}_0) \leq \frac{\ln 2}{4\kappa(2n\kappa+1)}.$$

*Then for all $k \geq 0$, we have*

$$0 \preceq \boldsymbol{G}_k - \nabla^2 f(\boldsymbol{x}_k) \preceq 2\kappa \left(1 - \frac{1}{n}\right)^k \mathrm{tr}\left(\nabla^2 f(\boldsymbol{x}_k)\right) \boldsymbol{I}_n,$$

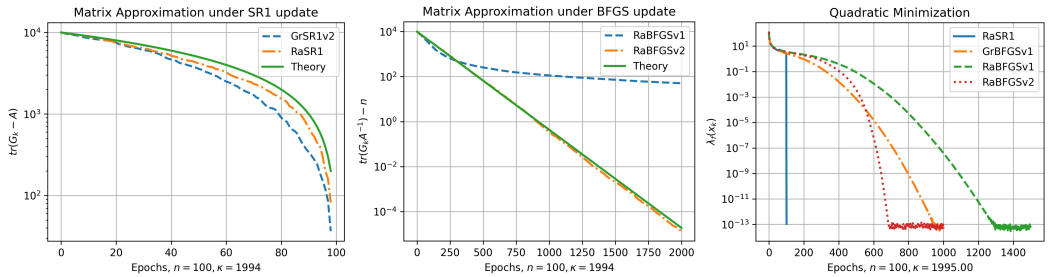
*and*

$$\lambda_f(\boldsymbol{x}_{k+1}) \leq 2n\kappa^2 \left(1 - \frac{1}{n}\right)^k \lambda_f(\boldsymbol{x}_k).$$

**Corollary 4.4** *Combining with [18, Theorem 4.7] (shown in Appendix G.1), if $\boldsymbol{x}_0$ satisfies $M\lambda_f(\boldsymbol{x}_0) \leq \frac{\ln \frac{3}{2}}{4\kappa}$, we obtain*

$$\lambda_f(\boldsymbol{x}_{k_0+k}) \leq \left(1 - \frac{1}{n}\right)^{k(k-1)/2} \left(\frac{1}{2}\right)^k \left(1 - \frac{1}{2\kappa}\right)^{k_0} \lambda_f(\boldsymbol{x}_0),$$

*with $k_0 \leq \max\{n, 2\kappa\} \ln(4n\kappa)$ for BFGS update and $k_0 \leq \max\{n, 2\kappa\} \ln(4n\kappa^2)$ for SR1 update.*




(a) (b) (c)

Figure 1: (a, b) Comparison of different direction choosing methods under SR1 or BFGS update for approximating $\boldsymbol{A}$ that $0 \prec \boldsymbol{A} \preceq L\boldsymbol{I}_n$ from $\boldsymbol{G}_0 = L\boldsymbol{I}_n$. (a) The variation of $\tau_{\boldsymbol{A}}(\boldsymbol{G}_k)$ during Random SR1 (RaSR1) and our Greedy SR1 (GrSR1v2) update with nearly matched upper bound. (b) The variation of $\sigma_{\boldsymbol{A}}(\boldsymbol{G}_k)$ during our Random BFGS (RaBFGSv2) update and the original random version (RaBFGSv1). (c) Comparison of SR1 and BFGS update for quadratic objective with the variation of $\lambda_f(\boldsymbol{x}_k)$. Here we only depict RaSR1 update, while the other SR1 methods share similar behavior.

**Remark 4.5** *Under SR1 update, unlike the measure $\sigma_{\boldsymbol{A}}(\cdot)$ used in [18], we employ a different measure $\tau_{\boldsymbol{A}}(\cdot)$, requiring a closer initial point condition to derive the convergence of $\lambda_k$. Fortunately, we could obtain the same convergence bound with slightly worse $k_0$.*

# 5 Numerical experiments

In this section, we verify our theorems through preliminary computational results for quasi-Newton methods. Rodomanov and Nesterov [18] have already compared the greedy quasi-Newton method they proposed with the classical quasi-Newton methods. They showed that GrDFP, GrBFGS, GrSR1 (greedy DFP, BFGS, SR1 methods) using directions based on $\bar{\boldsymbol{u}}_{\boldsymbol{A}}(\boldsymbol{G}) = \arg\max_{\boldsymbol{u} \in \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}} \frac{\boldsymbol{u}^\top \boldsymbol{G} \boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{A} \boldsymbol{u}}$, have quite competitive convergence with the standard one. They also presented the results for the randomized versions RaDFP, RaBFGS, RaSR1, which choose directions directly from a standard Euclidean sphere. They discovered that the randomized methods are slightly slower than the greedy ones. However, the difference is not really significant.

The difference between our algorithms and their methods mainly comes from the greedy strategy for SR1 and the random strategy for BFGS[3]. Hence, we mainly focus on exhibiting our validity in these schemes. We refer to GrSR1v2 as our revised method and GrSR1v1 as the previous method. Similarly, we denote RaBFGSv2 that uses scaled directions ($\boldsymbol{G}^{-1/2}\boldsymbol{u}$) and RaBFGSv1 that directly uses random directions correspondingly.

**Matrix approximation.** When using Algorithms 1 and 2 for approximating a matrix $\boldsymbol{A} \succ 0$, we show the measure as proved by our theorems 2.3 and 2.5 in Figures 1a and 1b. As Figure 1a depicts, our greedy and random SR1 updates (GrSR1v2 and RaSR1) share superlinear convergence rate under measure $\tau_{\boldsymbol{A}}(\cdot)$, while our theoretical bound matches them well. Moreover, Figure 1b describes the behavior of the random BFGS update. Note that the greedy BFGS update we proposed does not satisfy the $O(n^2)$ complexity, thus we leave it out. Our theory totally matches the linear convergence of measure $\sigma_{\boldsymbol{A}}(\cdot)$ in our modified random BFGS update (RaBFGSv2). However, directly choosing a direction from a spherically symmetric distribution (RaBFGSv1) fails to give such bounds. Clearly, our methods provide effective ways approaching a positive definite Hessian matrix.

**Quadratic minimization.** We also consider unconstrained quadratic minimization with the same positive definite matrix $\boldsymbol{A}$ and a randomly selected vector $\boldsymbol{b} \in \mathbb{R}^n$ from the standard Gaussian. Running Algorithm 3 with SR1 and BFGS updates, we all obtain the superlinear convergence of $\lambda_f(\cdot)$ as Theorem 3.2 shows in Figure 1c. Not surprisingly, our RaBFGSv2 runs faster than RaBFGSv1, while we also have the theoretical guarantee. At the same time, SR1-type methods converge to

[3]There is no difference in the random SR1 method compared to Rodomanov and Nesterov [18], which directly selects random directions. And our greedy BFGS method is not efficient as we mentioned in Remark 2.6, thus we leave it out.

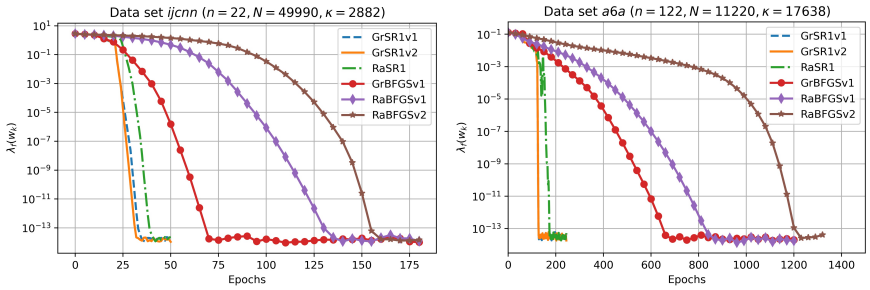



Figure 2: Comparison of SR1 and BFGS update for $\ell_2$-regularized logistic regression applied with 'ijcnn1' and 'a6a' data from the LIBSVM collection of real-world data sets. The regularized constant $\gamma = 1$, and $G_0 = LI_n$, where $L$ is the Lipschitz constant of the gradient. The lines of GrSR1v1 and GrSR1v2 in the right graph are totally overlapped.

zero after $n+1$ steps because $\boldsymbol{G}_n = \boldsymbol{A}$. Here, we only depict the RaSR1 update, while the other SR1 methods share the similar behavior. Although our theoretical bound can not directly match the experiments due to the related initial terms $\tau_{\boldsymbol{A}}(\boldsymbol{G}_0)$ and $\sigma_{\boldsymbol{A}}(\boldsymbol{G}_0)$, the decay terms: $(1-k/n)$ vs. $(1-1/n)^k$ already show the superiority of the SR1 update compared to the BFGS update in the scheme of quadratic minimization.

**Logistic regression.** At last, we consider the general objective following another popular test function, namely, $\ell_2$-regularized logistic regression:

$$f(\boldsymbol{w}) = \sum_{i=1}^{N} \ln\left(1 + e^{y_i \boldsymbol{w}^\top \boldsymbol{x}_i}\right) + \frac{\gamma}{2}\|\boldsymbol{w}\|^2, \ \boldsymbol{w} \in \mathbb{R}^n, \tag{19}$$

where $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^n$ are training samples with the corresponding labels $y_1, \ldots, y_N \in \{+1, -1\}$, and $\gamma > 0$ is the regularized coefficient. The Lipschitz constant of the gradient of $f$ can be taken as follows: $L = \frac{1}{4}\sum_{i=1}^{N}\|\boldsymbol{x}_i\|^2 + \gamma$. In particular, the Hessian-vector product for this function can be computed with the similar complexity of that for computing the gradient, guaranteeing $O(n^2)$ complexity in each step. We follow the same experiment design but we take data from the LIBSVM collection of real-world data sets for binary classification problems [6]. And we do not apply the *correction strategy* ($\tilde{\boldsymbol{G}}_k = (1 + Mr_k)\boldsymbol{G}_k$ in Algorithm 4) recommended by [18]. We always set $\gamma = 1$, a standard choice. In order to simulate the local convergence, we use the same initialization after running 5 standard Newton steps to make measure $\lambda_f(\boldsymbol{w}_0)$ small (around $0.1 \sim 10$).

The results are given in Figure 2. As we can see, the general picture is the same as for the previous objective. In particular, SR1-type methods are faster than BFGS-type methods, and greedy algorithms also converge more rapidly than random algorithms. The only difference is that our RaBFGSv2 has slower convergence behavior than RaBFGSv1. We consider our scaled direction is more suitable for a constant Hessian matrix as the quadratic objective has, thus we still have better convergence rate in the last few iterations that Hessians nearly unchanged in Figure 2. However, the Hessian varies drastically in the initial period, thus there is less benefit under a more accurate Hessian approximation. Overall, our proposed methods do not loose the superlinear convergence rate in large condition number schemes, while we also present the theoretical guarantee for these random algorithms.

## 6 Conclusion

In this work we have studied the behavior of two famous quasi-Newton methods: the SR1 and the BFGS methods. We have presented different greedy methods in contrast to Rodomanov and Nesterov [18], and the random version of these methods. In particular, we have provided the faster Hessian approximation behavior and the condition-number-free (local) superlinear convergence rates applied to quadratic or strongly self-concordant objectives. Moreover, the experiments corroborate our analysis well. Note that our current results do not give the superlinear convergence rate of the standard quasi-Newton methods with only gradient information. However, we hope that the theoretical analysis and the related work will be useful for understanding theoretically quasi-Newton methods in a more specific view.

# References


[1] Charles G Broyden. Quasi-newton methods and their application to function minimisation. *Mathematics of Computation*, 21(99):368–381, 1967.

[2] Charles G Broyden. The convergence of a class of double-rank minimization algorithms: 2. the new algorithm. *IMA journal of applied mathematics*, 6(3):222–231, 1970.

[3] Charles George Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

[4] Charles George Broyden, John E Dennis Jr, and Jorge J Moré. On the local and superlinear convergence of quasi-newton methods. *IMA Journal of Applied Mathematics*, 12(3):223–245, 1973.

[5] Richard H Byrd, Jorge Nocedal, and Ya-Xiang Yuan. Global convergence of a cass of quasi-newton methods on convex problems. *SIAM Journal on Numerical Analysis*, 24(5):1171–1190, 1987.

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

[7] William C Davidon. Variable metric method for minimization. *SIAM Journal on Optimization*, 1(1):1–17, 1991.

[8] John E Dennis and Jorge J Moré. A characterization of superlinear convergence and its application to quasi-newton methods. *Mathematics of computation*, 28(126):549–560, 1974.

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

[10] Roger Fletcher and Michael JD Powell. A rapidly convergent descent method for minimization. *The computer journal*, 6(2):163–168, 1963.

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

[12] Gene H Golub and Charles F Van Loan. Matrix computations, 1996.

[13] Andreas Griewank and Ph L Toint. Local convergence analysis for partitioned quasi-newton updates. *Numerische Mathematik*, 39(3):429–448, 1982.

[14] Dmitry Kovalev, Robert M Gower, Peter Richtárik, and Alexander Rogozin. Fast linear convergence of randomized bfgs. *arXiv preprint arXiv:2002.11337*, 2020.

[15] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

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

[17] Anton Rodomanov and Yurii Nesterov. New results on superlinear convergence of classical quasi-newton methods. *Journal of Optimization Theory and Applications*, 188(3):744–769, Jan 2021. ISSN 1573-2878. doi: 10.1007/s10957-020-01805-8. URL `http://dx.doi.org/10.1007/s10957-020-01805-8`.

[18] Anton Rodomanov and Yurii Nesterov. Greedy quasi-newton methods with explicit superlinear convergence. *SIAM Journal on Optimization*, 31(1):785–811, 2021.

[19] Anton Rodomanov and Yurii Nesterov. Rates of superlinear convergence for classical quasi-newton methods. *Mathematical Programming*, pages 1–32, 2021.

[20] David F Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.

[21] Andrzej Stachurski. Superlinear convergence of broyden's bounded $\theta$-class of methods. *Mathematical Programming*, 20(1):196–212, 1981.

[22] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[23] Hiroshi Yabe and Naokazu Yamaki. Local and superlinear convergence of structured quasi-newton methods for nonlinear optimization. *Journal of the Operations Research Society of Japan*, 39(4):541–557, 1996.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See conclusion in Section 6. The main limitation is that our current results do not give the superlinear convergence rate of the standard quasi-Newton methods with only gradient information. Our proposed greedy and random quasi-Newton methods follow Rodomanov and Nesterov [18]'s work, and need to employ Hessian-vector product, though totally complexity is $O(n^2)$ as standard quasi-Newton methods.
   (c) Did you discuss any potential negative societal impacts of your work? [No] The purpose of this work is to provide a better understanding of quasi-Newton methods.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] The assumptions mainly include: strongly self-concordant (Definition 4.1), smooth and strongly-convex objective.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix for detail.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code are proprietary, but we will dispaly the URL after we clear up.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5 for detail.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We wants to compare the traing dynamic, and different trials may cause different nemerical results, which can not be observed clearly in one graph.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] The experiments is certainly simple and easy to run under CPUs.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A] These datasets are common in works.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] These datasets are common in works.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] These datasets are common in works.
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]