Convergence Rate of Greedy SR1 With Trust Region Method

Yu Xia * and Hao Wu

School of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing, China

* Corresponding Author Email: nuaaxiayu@nuaa.edu.cn

Abstract. Recently, Greedy Quasi-Newton methods have attracted wide interests of some researchers for their explicit superlinear convergence rate. These algorithms which achieve rapid local convergence construct a matrix sequence to approximate the Hessian matrix of the objective function iteratively. This paper proposes an algorithm, GR-SR1-TR, which incorporate Greedy SR1 method with Trust Region framework and employs a new correction technique. We prove that the approximation matrix sequence has a linear descent property under the Frobenius-norm metric. Further, both the global convergence and an explicit superlinear rate are established. The effectiveness of GR-SR1-TR has been verified by preliminary numerical experiments.

Keywords: superlinear convergence; Quasi-Newton method; SR1 method; Trust Region method.

1. Introduction

1.1. Background

In this paper, we consider the unconstrained minimization problem as shown below

$$\min_{x\in R^n} f(x), \tag{1}$$

Where the function is strongly convex and its gradient is Lipschitz continuous. Quasi-Newton methods are important optimization algorithms for solving unconstrained optimization problems. They use approximation matrix instead of the Hessian matrix, compared to the Newton method. Quasi-Newton methods, including the DFP algorithm [1,2], the BFGS algorithm [1,3,4,5] and the SR1 algorithm [1,6] have received a lot of attention on its global convergence [7,8,9,10,11,12] and local superlinear convergence [13,14,15,16]. Researches in the 1970s [17,18,19,20,21] have demonstrated that these Quasi-Newton methods have a asymptotic superlinear convergence rate, the ratio of successive residuals tends to zero as the iterations approach to infinity, i.e.

$$\lim_{k\to+\infty} \frac{\|x_{k+1}-x_*\|}{\|x_k-x_*\|} = 0, \tag{2}$$

Where $\{x_k\}$ is the iterative approximate sequence of the optimal solution, k is the number of iterations, and x_* is the optimal solution. However, it's unknown whether the residual converges like $O(c^{k^2})$ or $O(k^{-k})$, where $c\in(0,1)$ is a constant. Therefore, the theory still remains insufficient, and a specific superlinear convergence rate has yet to be established.

Although there have been many analyses on the asymptotic convergence rate of Quasi-Newton methods, these analyses lack an explicit analysis to accurately describe the convergence process. In recent years, Rodomanov and Nesterov proposed Greedy Quasi-Newton methods [22], which are variants of classical Quasi-Newton methods.

It's necessary to briefly introduce the idea of Greedy Quasi-Newton methods. These algorithms construct a matrix sequence which approximates the Hessian matrix of the objective function iteratively, aiming to achieve rapid local convergence. To construct a matrix sequence that approximates the Hessian matrix, greedy technique has been proposed, which involves greedily maximizing the difference between $\sigma_G(B)$ and $\sigma_G(B_+)$, were

$$B_{+} = B - \frac{(B-G)\bar{s}\,\bar{s}^{\mathrm{T}}(B-G)}{\bar{s}^{\mathrm{T}}(B-G)\bar{s}}, \tag{3}$$

$$\sigma_G(B) \stackrel{def}{=} Tr[(B-G)G^{-1}], \tag{4}$$

$$\sigma_G(B) - \sigma_G(B_{+}) \geq \frac{\langle (B-G)s, s\rangle}{\langle Gs, s\rangle}, \tag{5}$$

G is the Hessian matrix, B is called approximation matrix, s^{T} denotes the transpose of s, and (3) is the update formula of the Greedy SR1 method. Rodomanov and Nesterov maximize the lower bound in the right side of (5) to maximize the difference between $\sigma_G(B)$ and $\sigma_G(B_{+})$.

However, the cost of solving this maximization problem is enormous. Instead of directly solving the maximum solution, they choose to use a vector which reach the maximum value in a basis, with the specific step shown in (6),

$$\bar{s} \in \arg\max_{s\in\{e_1,\ldots,e_n\}} \frac{\langle (B-G)s, s\rangle}{\langle Gs, s\rangle}, \tag{6}$$

where $\langle ,\rangle$ is inner product operation, $\bar{s}$ is called the greedy direction from B to G, e_i is the vector whose $i-th$ element is one while the other elements are zero. It should be noted that (5) is not always holds. Rodomanov and Nesterov point out a specifical conditions to ensure (5) hold [22], which is shown below,

$$B - G \succeq 0. \tag{7}$$

To ensure (7) hold, they use a correction technique

$$\tilde{B}_k = (1+\delta r_k)B_k, \tag{8}$$

where B_k is the $k-th$ approximation matrix, $r_k = \|x_{k+1} - x_k\|_{x_k}$, δ is a sufficiently large number.

In their works, they elaborate on the explicit local superlinear convergence rate of Greedy Quasi-Newton methods with a fixed step size and obtain the rate of the form like

$$(1-\frac{1}{n\kappa})^{k(k-1)/2}(\frac{1}{2})^{k}(1-\frac{1}{2\kappa})^{k_0}, \tag{9}$$

where n is the dimension of parameters, κ is the condition number of the objective function, k_0 is the iteration number where superlinear rate starts from and k is the iteration number after k_0. In addition, they showed the Greedy SR1 method can reach faster rate on the approximation of Hessian matrix than the other methods in the Broyden family. Subsequently, some researchers proposed a new Greedy SR1 method with tighter upper bounds [23].

Rodomanov and Nesterov also derived the explicit superlinear convergence rate of classical Quasi-Newton methods in the convex Broyden family under specific conditions [24], including the BFGS method and the DFP method. Inspired by Rodomanov and Nesterov, some scholars have proposed modified SR1 methods with explicit convergence rates [25]. In addition, some other scholars have demonstrated explicit superlinear convergence rate of the Quasi-Newton method within the convex Broyden family in the face of different objective functions [26]. Lin et al. proved theoretical results for randomized greedy directions and obtained faster explicit convergence results by employing a different way to select greedy directions [23]. Recently, researchers combined the greedy PSB method and the Trust Region framework and proved its explicit superlinear convergence without using correction technique [27].

1.2. Contributions

We prove that the approximation matrix sequence has a linear descent property under the Frobenius-norm metric. Then we propose an algorithm, GR-SR1-TR, which incorporates Greedy SR1 method with Trust Region framework and employ a new correction technique, which reduces the influence of correction. Our algorithm not only achieves global convergence but also maintains the explicit convergence rate under specific conditions.

1.3. Notations

For a smooth function, $\nabla f(x), \nabla^2 f(x)$ are its gradient and Hessian matrix respectively. Furthermore, the following notations will be used in the proofs of this paper,

$$\|h\|_A = \langle Ah, h\rangle^{1/2} \tag{10}$$

$$\|s\|_A^* = \langle s, A^{-1}s\rangle^{1/2} \tag{11}$$

$$\lambda_f(x) = \|\nabla f(x)\|_x^* = \langle \nabla f(x), \nabla^2 f(x)^{-1}\nabla f(x)\rangle^{1/2} \tag{12}$$

when A is the Hessian matrix at x, they are abbreviated as $\|h\|_x$ and $\|s\|_x^*$. $\langle,\rangle$ is inner product operation, $\lambda = \lambda_f(x)$, and x_+ means the next iteration of x.

2. Linear convergence of matrix differences

In this section, we will describe the process of iterations about the Greedy SR1 method under the Frobenius-norm metric. First, we need to recall the general update form of the classical SR1 method and its greedy form. The general update formula of the classical SR1 method is shown in (13),

$$SR1(G, B, s) = B - \frac{(B-G)ss^{\mathrm{T}}(B-G)}{s^{\mathrm{T}}(B-G)s}, \tag{13}$$

if $s^{\mathrm{T}}(B-G)s = 0$, then we set $SR1(G,B,s) = B$, where G is called the target matrix, B is called the approximation matrix, s is called the approximation direction from B to G, $SR1(G,B,s)$ is called the update matrix, and B^{T} denotes the transpose of matrix B. In the classical SR1 method, the target matrix G in the $k-th$ iteration is taken to be the matrix J_k given by (14),

$$J_k = \int_0^1 \nabla^2 f(x_{k+1} + ts_k)dt, \tag{14}$$

where $s_k = x_{k+1} - x_k$. J_k does not need to be solved separately in practical due to the Quasi-Newton equation $y_k = J_k s_k$, where $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, then we have

$$SR1(B_k, s_k, y_k) = B_k - \frac{(B_k s_k - y_k)(B_k s_k - y_k)^{\mathrm{T}}}{(s_k)^{\mathrm{T}}(B_k s_k - y_k)} \tag{15}$$

Although the classical SR1 method can approximate the Hessian matrix at the optimum point along the direction of the cut line $d_k = x_k - x_{k-1}$ (the approximation direction s in classical Quasi-Newton methods is d_k), i.e.

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(x_*))d_k\|_2}{\|d_k\|_2} = 0, \tag{16}$$

it does not have a comprehensive approximation for any direction s. Thus, it cannot achieve a local fast quadratic convergence rate like the Newton method.

In contrast, the Greedy SR1 method approximate the Hessian matrix iteratively by using greedy direction in (6), which achieves fast local convergence. Nestrov and Rodomanov first proposed Greedy Quasi-Newton methods by greedily selecting a specific approximation direction s in a basis to maximize the difference between two adjacent iterations with regard to $\sigma_G(B) = Tr[(B-G)G^{-1}]$ (defined in (4)), as showed in (5) and (6).

Furthermore, the analyses in these studies rely on the use of a fixed step size. However, in the Trust Region framework, which is usually used for the SR1 method, the iterates are not directly generated by the iterative formula at each step. The iterates of the SR1 method with Trust Region framework are obtained by solving a subproblem, which cannot travel in a given direction according to a fixed step size. Thus, no explicit rate can be obtained directly by following the method in [22,23]. In contrast, Ji and Dai constructed the explicit rates under the Trust Region framework for greedy PSB method. In their works, they select Frobenius-norm as the measure of difference between B and G under the Trust Region framework and their method does not rely on a fixed step size under the Trust Region framework. Therefore, we follow this thought which adopts Frobenius-norm as the measure in this paper.

The following lemma shows in the first place that the difference of the approximation matrix B with respect to the target matrix G is decreasing in a specific metric.

Theorem 2.1: *For the SR*1 *method, if the approximation matrix B and the target matrix G satisfy $B \succeq G$, then for any approximation direction s, there has*

$$\left\|SR1(G,B,s)-G\right\|_F^2 \le \left\|(B-G)\right\|_F^2 - \left[\frac{s^{\mathrm{T}}(B-G)(B-G)s}{s^{\mathrm{T}}(B-G)s}\right]^2. \tag{17}$$

Proof: Expand the equation $\left\|SR1(G,B,s)-G\right\|_F^2$ as below,

$$\left\|SR1(G,B,s)-G\right\|_F^2 = \left\|(B-G)\right\|_F^2 - 2[\frac{s^{\mathrm{T}}(B-G)(B-G)(B-G)s}{s^{\mathrm{T}}(B-G)s}] + [\frac{s^{\mathrm{T}}(B-G)(B-G)ss^{\mathrm{T}}(B-G)(B-G)s}{s^{\mathrm{T}}(B-G)ss^{\mathrm{T}}(B-G)s}],$$

By combining $B \succeq G$ with the Cauchy-Schwartz inequality, we can obtain (18),

$$t^{\mathrm{T}}(B-G)t/(s^{\mathrm{T}}t) \ge (t^{\mathrm{T}}t)^2/(s^{\mathrm{T}}t)^2, \tag{18}$$

where $t=(B-G)s$, hence

$$\left\|SR1(G,B,s)-G\right\|_F^2 \le \left\|(B-G)\right\|_F^2 - \left[\frac{s^{\mathrm{T}}(B-G)(B-G)s}{s^{\mathrm{T}}(B-G)s}\right]^2. \qquad \square$$

Maximize the difference between $\left\|SR1(G,B,s)-G\right\|_F^2$ and $\left\|(B-G)\right\|_F^2$ now, when $s^{\mathrm{T}}(B-G)s \neq 0$, we propose a novel greedy direction as follows,

$$\bar{s}_G(B) \in \arg\max_{s\in\{e_1,\ldots,e_n\}} \frac{\left\|(B-G)s\right\|_2^2}{s^{\mathrm{T}}(B-G)s}, \tag{19}$$

where e_i is the vector whose $i-th$ element is one and the other elements are zero. (19) picks approximate directions in the basis to maximize the descent in (17), then we can obtain the greedy direction. With (19), we have the update formula

$$SR1(G,B,\bar{s}_G(B)) = B - \frac{(B-G)\bar{s}_G(B)\bar{s}_G(B)^{\mathrm{T}}(B-G)}{\bar{s}_G(B)^{\mathrm{T}}(B-G)\bar{s}_G(B)}. \tag{20}$$

It is worth noting that the distinction between greedy directions in [22,23] and our paper lies only in the difference about the proving process and theoretical results.

The following theorem elaborates that when we take the approximation direction s as the greedy direction defined by (19), the approximation matrix sequence exhibits a linear descent property under the Frobenius-norm metric.

Theorem 2.2: *Consider the update* (20) *for the Greedy SR*1 *method, if* $B \succeq G$ *holds, we have*

$$\left\|SR1(G,B,s)-G\right\|_F \le \sqrt{1-\frac{1}{n}}\left\|B-G\right\|_F . \tag{21}$$

Proof:

$$\left[\frac{\bar{s}_G(B)^{\mathrm{T}}(B-G)(B-G)\bar{s}_G(B)}{\bar{s}_G(B)^{\mathrm{T}}(B-G)\bar{s}_G(B)}\right]^2 \overset{(19)}{\ge} \frac{1}{n}\sum_{i=1}^{n}\frac{e_i^{\mathrm{T}}(B-G)(B-G)e_ie_i^{\mathrm{T}}(B-G)(B-G)e_i}{e_i^{\mathrm{T}}(B-G)e_ie_i^{\mathrm{T}}(B-G)e_i}$$

$$=\frac{1}{n}\sum_{i=1}^{n}trace[\frac{e_i^{\mathrm{T}}(B-G)(B-G)e_ie_i^{\mathrm{T}}(B-G)(B-G)e_i}{e_i^{\mathrm{T}}(B-G)e_ie_i^{\mathrm{T}}(B-G)e_i}]$$

$$=\frac{1}{n}trace[(B-G)(B-G)\sum_{i=1}^{n}\frac{e_ie_i^{\mathrm{T}}(B-G)(B-G)e_ie_i^{\mathrm{T}}}{e_i^{\mathrm{T}}(B-G)e_ie_i^{\mathrm{T}}(B-G)e_i}] .$$

Let

$$\sum_{i=1}^{n}\frac{e_ie_i^{\mathrm{T}}(B-G)(B-G)e_ie_i^{\mathrm{T}}}{e_i^{\mathrm{T}}(B-G)e_ie_i^{\mathrm{T}}(B-G)e_i}=P ,$$

where P is a diagonal matrix which satisfies $P \succeq I$. Also, since $(B-G)(B-G)$ has positive diagonal elements, we have that

$$\frac{1}{n}trace[(B-G)(B-G)P] \ge \frac{1}{n}trace[(B-G)(B-G)] . \tag{22}$$

Hence, when $s=\bar{s}_G(B)$, we have

$$\left\|SR1(G,B,\bar{s}_G(B))-G\right\|_F^2 \le (1-\frac{1}{n})\left\|B-G\right\|_F^2 ,$$

that is

$$\left\|SR1(G,B,s)-G\right\|_F \le \sqrt{1-\frac{1}{n}}\left\|B-G\right\|_F .$$

3. Greedy SR1 algorithm with Trust Region

Global convergence has not been established for Greedy Quasi-Newton methods with the fixed step. Besides, its local convergence as well as explicit rate all depend on the correction technique $\tilde{B}_k=(1+\delta r_k)B_k$. The correction technique is showed in (8) previously, which ensures $B-G\succeq 0$, where δ is a sufficiently large number and $r_k=\left\|x_{k+1}-x_k\right\|_{x_k}$.

In previous works [22,23], to guarantee the convergence and obtain explicit convergence rate, the specific correction coefficient δ need to be figured out. However, this correction method has two drawbacks. First, there are some functions for which no specific coefficients can be determined. Secondly, the coefficients δ in previous correction technique make a large bias to the approximation matrix and then affect the rate of the practical algorithm.

Besides, when one refers to the global convergence of the SR1 method, the most widely used is the Trust Region framework. The SR1 method globally convergent under some appropriate conditions in the Trust Region framework[28]. Thus, it is necessary to introduce some notations of the Quasi-Newton method under the Trust Region framework first,

$$\min q(d)= f(x_k)+\nabla f(x_k+d)^{\mathrm{T}}d+\frac{1}{2}d^{\mathrm{T}}B_kd \tag{23}$$

$$s.t.\left\|d\right\|_2 \le \Delta_k \tag{24}$$

We call (23) and (24) the Trust Region subproblem and $q(d)$ is the quadratic model of this subproblem. Δ_k is the Trust Region radius and B_k is the approximation matrix of the Hessian matrix. Then we introduce the trust ratio, which adjusts the size of the Trust Region radius in the next iteration,

$$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{q^{(k)}(0) - q^{(k)}(d_k)} . \tag{25}$$

To ensure the global convergence and address the problem in previous correction technique, we propose our algorithm in this section, called GR-SR1-TR, which achieve the global convergence while maintain local superlinear convergence as well as explicit convergence rate.

Some reasonable assumptions are made on the objective function, which assume that both its gradient and Hessian matrix are Lipschitz continuous as well as the function is strong convex with γ, as indicated below,

$\|\nabla f(x) - \nabla f(y)\|_F \preceq L\|x - y\|$,

$\|\nabla^2 f(x) - \nabla^2 f(y)\|_F \preceq L\|x - y\|$,

$f(y) \geq f(x) + \nabla f(x)^{\mathrm{T}}(y - x) + \gamma/2\|y - x\|^2$.

The algorithm, GR-SR1-TR, is shown below, which is set in the Trust Region framework. Subsequently, we note some instructions and demonstrate the convergence property as well as convergence rate.

Algorithm: GR-SR1-TR

Greedy SR1 with the Trust Region method(GR-SR1-TR)

1.Initialization:

select x_0, $\Delta\max$, Δ_0, $0 < \eta_1 < \eta_2 < \eta_3 < 1$, $0 < \gamma_1 < 1 < \gamma_2$, and set $k = 0$, $B_0 = LI$, $\delta = \delta_0$.

2.Iteration: for $k \geq 0$, do

2.1 stop the iteration when the termination criteria is satisfied.

2.2 solving subproblems,

$$\min q(d) = f(x_k) + \nabla f(x_k + d)^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} B_k d ,$$

then compute $f(x_k + d_k)$ and ρ_k.

2.3 update iterates,

$$x_{k+1} = \begin{cases} x_k + d_k, if\ \rho_k \geq \eta_1, \\ x_k, \quad otherwise. \end{cases}$$

2.4 update radius,

$$\Delta_{k+1} = \begin{cases} \gamma_1 \Delta_k, if\ \rho_k < \eta_2, \\ \gamma_2 \Delta_k, if\ \rho_k \geq \eta_3, and\ \|d_k\| \geq \Delta_k, \\ \Delta_k, \quad otherwise. \end{cases}$$

2.5 update matrix, let $\tilde{B} = (1 + \delta r_k) B_k$, where $r_k = \|x_{k+1} - x_k\|_{x_k}$,

if $t^{\mathrm{T}}(\tilde{B} - G)t / (\bar{s}^{\mathrm{T}} t) \geq (t^{\mathrm{T}} t)^2 / (\bar{s}^{\mathrm{T}} t)^2$, then

$$B_{k+1} = SR1(G_k, \tilde{B}, \bar{s}) ,$$

where $G_k = \nabla^2 f(x_{k+1})$, $t = (\tilde{B} - G)\bar{s}$ and

$$\bar{s} = \underset{s \in \{e_1, \ldots, e_n\}}{\arg\max} \frac{\|(\tilde{B} - G_k)s\|_2^2}{s^{\mathrm{T}}(\tilde{B} - G_k)s},$$

else $\delta = \rho\delta$, and go to 2.5.

3. $k = k + 1$ and go to step 2.

It is necessary to clarify that: 1) the subproblem does not need to be solved exactly, it is enough to satisfy (26), which makes it possible to analysis the inexact solution,

$$q^{(k)}(0)-q^{(k)}(d_k)\geq \beta_1\|g_k\|_2 \min\left\{\Delta_k,\frac{\|g_k\|_2}{\|B_k\|_2}\right\},\beta_1\in(0,1]. \tag{26}$$

When $\beta_1=0.5$, it is the descent of the Cauchy point [29]; 2) we choose the initial matrix $B_0=LI$ to make the initial situation satisfy the discriminant condition, in addition the parameters in our algorithm are not sensitive; 3) for other Quasi-Newton methods, the algorithm is unable to choose the appropriate initial point which satisfies the discriminant condition, then it may lead to slower practical convergence in the initial stage and the other Quasi-Newton methods lack the necessary theoretical proofs in the Trust Region framework; 4) there exist a certain δ which is not need to be figured out, which satisfy (18) in the proof of Theorem 2.1, i.e. the condition as below,

$t^{\mathrm{T}}(\tilde{B}-G)t/(\bar{s}^{\mathrm{T}}t)\geq(t^{\mathrm{T}}t)^2/(\bar{s}^{\mathrm{T}}t)^2$.

Then we demonstrate the convergence property as well as convergence rate.

Theorem3.1: *The iterates are generated by the algorithm (GR-SR1-TR), then for each iteration, there exists a constant δ ensuring* (18) *hold.*

Proof: There exists a constant $\tilde{l}$ as below,

$\tilde{l}=\log_{\rho}^{\frac{M}{\delta_0}}$,

which satisfies $\delta_0\rho^{\tilde{l}}\geq M$, where $\rho\geq 1$. Then there holds

$\tilde{B}\succeq G$.

Then, theorem 3.1 holds from theorem 2.1 and theorem 2.2. □

The following establishes the global convergence of the algorithm under certain conditions.

Lemma3.1[29]**:** *The Quasi-Newton method with the Trust Region framework is globally convergent if the function is lower bounded on the lower level set, its gradient is Lipschitz continuous in an open neighborhood of the lower level set and the sequence of matrices for the subproblems of the Trust Region method is bounded and satisfies the* (26) *with $\eta_1>0$.*

Corollary3.1: *If both the gradient and the Hessian matrix of the objective function are Lipschitz continuous as well as the objective function is strongly convex, i.e., satisfy*

$$\nabla^2 f(x_k)\preceq LI, \tag{27}$$

$$\left\|\nabla^2 f(x)-\nabla^2 f(y)\right\|_F\preceq L\|x-y\|, \tag{28}$$

then the algorithm (GR-SR1-TR) converges globally when the solution of the Trust Region subproblem of the algorithm satisfies (26)*.*

After verifying the global convergence of our algorithm, we proceed to analyze its convergence rate. First, we need to confirm that it achieves the local asymptotic superlinear convergence rate. An immediate conclusion of Corollary 3.1 will be given to proof this convergence rate.

Corollary3.2: *The iterates x_k are given by the algorithm (GR-SR1-TR) and the algorithm satisfies conditions in Theorem* 3.1*, then*

$$\lim_{k\to\infty}(B_k-G_k)=0 \quad . \tag{29}$$

Convergence rate of the Quasi-Newton method in the Trust Region framework depends on the degree of proximity to the Hessian matrix sequence. We state this fact by Theorem 3.2.

Theorem3.2[27]**:** *If the Quasi-Newton method with the Trust Region framework satisfies the conditions in Corollary* 3.1 *and satisfies* (29) *, if* (30) *is satisfied when the subproblem is taken at the boundary of the Trust Region,*

$$q^{(k)}(\alpha d_k)\geq q^{(k)}(d_k),\forall\alpha\in[0,1], \tag{30}$$

or (3.9) *is satisfied when the subproblem is taken in the Trust Region radius,*

$$d_k=-B_k^{-1}g_k, \tag{31}$$

then it converges superlinearly.

Corollary3.3: *The iterates* x_k *are given by the algorithm (GR-SR1-TR) and the algorithm satisfies the conditions in Theorem* 3.1*, then the algorithm converges superlinearly.*

Before exploring the rate of explicit convergence, as previous scholars do, we need to make certain assumptions about the bound on second-order information of the objective function, i.e., there exists a constant $M \geq 0$ such that for any vector x, y, z, w, there has

$$\nabla^2 f(y) - \nabla^2 f(x) \preceq M \|y - x\|_z \nabla^2 f(w). \tag{32}$$

This assumption requires that the Hessian matrices cannot differ too much from each other. We call the function satisfying (32) a strongly self-concordant function, where the constant M is called strong self-concordant coefficient.

Lemma3.2[27]**:** *If there exist constants* θ_1, θ_2 *satisfying*

$0 < \theta_1 \leq 1 \leq \theta_2 < 1/\theta_1$,

such that

$\theta_1 \nabla^2 f(x) \preceq B \preceq \theta_2 \nabla^2 f(x)$,

then we have

$$\|x_+ - x\|_x \leq \frac{\lambda}{\theta_1} \tag{33}$$

and

$$\lambda_f(x_+) \leq (1 + \frac{M\lambda}{2\theta_1}) \frac{1 - \theta_1 + \frac{M\lambda}{2\theta_1}}{\theta_1} \lambda, \tag{34}$$

were

$\lambda = \lambda_f(x), x_+ = x - B^{-1} \nabla f(x)$.

It should be noted that the strong self-concordant coefficient in this paper only plays the role of boundedness, and therefore do not require the determination of specific value as in the case of previous correction method. Up to this point, the algorithm can be analyzed for explicit upper bounds when the strong self-concordant function act as the objective function.

Theorem3.3: *The iterates* x_k *are given by the algorithm (GR-SR1-TR) and the algorithm satisfies conditions in Theorem* 3.1*, then*

$$\lambda_f(x_k) \leq \mathrm{O}(\sqrt{1 - \frac{1}{n}}^k) \lambda_f(x_{k-1}). \tag{35}$$

Proof: First, we define a constant

$\alpha \overset{def}{=} \frac{1}{2}\sqrt{1 - \frac{1}{n}}$.

By theorem 3.2, it follows that $\{d_k\}$ converges to zero superlinearly. Then, for constant α there exists $k_0 \geq 0$ satisfies $\|d_k\|_2 \leq \alpha^k$ for any $k \geq k_0$, and $\alpha^{k_0} \leq 1/2$. So, we have

$$L \sum_{i=k_0}^{k-1} (\|d_i\| \beta^{k-i}) \leq L \sum_{i=k_0}^{k-1} (\alpha^i \beta^{k-i}) \leq \frac{L\alpha^{k_0}}{1 - \frac{\alpha}{\beta}} \beta^{k-k_0} \leq L\beta^{k-k_0}. \tag{36}$$

Since that

$\left\|B_k - \nabla^2 f(x_k)\right\|_F \leq \beta^{(k-1)} \left\|B_{k-1} - \nabla^2 f(x_k)\right\|_F$

$\leq \beta^{(k-1)} (\left\|B_{k-1} - \nabla^2 f(x_{k-1})\right\|_F + \left\|\nabla^2 f(x_{k-1}) - \nabla^2 f(x_k)\right\|_F)$

$$\leq \beta^{(k-1)}\left\|B_{k-1}-\nabla^2 f(x_{k-1})\right\|_F+\beta^{(k-1)}L\left\|d_{k-1}\right\|$$

$$\leq\cdots\leq\prod_{i=0}^{k-1}\beta^{(i)}\left\|B_0-G_0\right\|_F+L\sum_{j=0}^{k-1}(\left\|d_j\right\|\prod_{i=j}^{k-1}\beta^{(i)}) \tag{37}$$

$$=\sqrt{1-\frac{1}{n}}^{k}\left\|B_0-G_0\right\|_F+L\sum_{j=0}^{k-1}(\left\|d_j\right\|\prod_{i=j}^{k-1}\beta^{(i)}), \tag{38}$$

for any $k\geq k_0$ there holds

$$\left\|B_k-G_k\right\|_F\leq\sqrt{1-\frac{1}{n}}^{k-k_0}\left\|B_0-G_0\right\|_F+L\sum_{I=k_0}^{k-1}(\left\|d_j\right\|\beta^{k-i}). \tag{39}$$

From (36) and (37), for any $k\geq k_0$,

$$\left\|B_k-G_k\right\|_F\leq(L+\left\|B_{k_0}-G_{k_0}\right\|_F)\beta^{k-k_0}. \tag{40}$$

Let

$\underline{\theta}_k=1-\frac{1}{\mu}(L+\left\|B_{k_0}-G_{k_0}\right\|_F)\beta^{k-k_0}$, $\bar{\theta}_k=1+\frac{1}{\mu}(L+\left\|B_{k_0}-G_{k_0}\right\|_F)\beta^{k-k_0}$,

from (40) and $G_k\succeq\mu I$, we get

$$\bar{\theta}_k G_k\succeq B\succeq\underline{\theta}_k G_k. \tag{41}$$

And since $\{d_k\}$ converges superlinearly to zero can be obtained by $\{\lambda_f(x_k)\}$ converging superlinearly to zero, so for sufficiently large k, there hold

$$(1+\frac{M\lambda_f(x_k)}{2\underline{\theta}_k})\frac{1}{\underline{\theta}_k}\leq 2 \tag{42}$$

and

$$1-\underline{\theta}_k+\frac{M\lambda_f(x_k)}{2\underline{\theta}_k}\leq\frac{2}{\mu}(L+\left\|B_{k_0}-G_{k_0}\right\|_F)\beta^{k-k_0}. \tag{43}$$

From (42) and (43), for sufficiently large k we have

$$(1+\frac{M\lambda_f(x_k)}{2\underline{\theta}_k})\frac{1-\underline{\theta}_k+\frac{M\lambda_f(x_k)}{2\underline{\theta}_k}}{\underline{\theta}_k}\leq\frac{4}{\mu}(L+\left\|B_{k_0}-G_{k_0}\right\|_F)\beta^{k-k_0}. \tag{44}$$

Then by Lemma 3.2 and (44), we can prove (35). □

We have thus obtained the explicit convergence rate of the algorithm (GR-SR1-TR) for strong self-concordant functions. When the object function is not a strong self-concordant function, our algorithm still remains convergent, owning to convergence properties of the Trust Region method [29], while the previous correction technique [22,23] may fail to obtain a convergence guarantee due to the inability to determine the strong self-concordant coefficients M.

4. Numerical experiment

In the experimental part, our algorithm and the existing Greedy Quasi-Newton algorithms are compared regarding the number of iterations and running time on MATLAB. We consider minimizing the log-sum-exp function

$$f(x)=\ln(\sum_{j=1}^{m}e^{\langle c_j,x\rangle-b_j})+\frac{1}{2}\sum_{j=1}^{m}\langle c_j,x\rangle^2+\frac{1}{2}\|x\|^2. \tag{45}$$

This function is one of strong self-concordant functions. Besides, its gradient is Lipschitz continuous with L and the function is strong convex. Its Lipschitz constant on gradient is

$$L=2\sum_{j=1}^{m}\|c_j\|^2+1,$$

and its strong self-concordant coefficient is M =2 (see more details in[22]). The data c_j, b_j in the function are generated in the same way as in the previous article [22,27].

In Tables 1-5, the variation about the norm of gradient with respect to the number of iterations is shown for different parameters. The variation about the norm of gradient with respect to time is shown in Tables 5-10 for different parameters as well. GR-SR1$_1$ represents the algorithm in [22], GR-SR1$_2$ represents the algorithm in [23], GR-PSB-TR represents the Greedy PSB method with Trust Region in [27] and GR-SR1-TR is our algorithm. ε refers to the accuracy at which algorithms reach.

Table 1. (n=m=50)

ε	GR-SR1$_1$	GR-SR1$_2$	GR-PSB-TR	GR-SR1-TR
10^{-1}	37	37	34	32
10^{-3}	58	58	51	51
10^{-5}	67	68	51	52
10^{-7}	74	74	52	53
10^{-9}	80	80	52	53

Table 2. (n=m=150)

ε	GR-SR1$_1$	GR-SR1$_2$	GR-PSB-TR	GR-SR1-TR
10^{-1}	91	90	89	89
10^{-3}	160	160	151	150
10^{-5}	186	186	151	151
10^{-7}	205	205	153	152
10^{-9}	218	217	153	152

Table 3. (n=m=250)

ε	GR-SR1$_1$	GR-SR1$_2$	GR-PSB-TR	GR-SR1-TR
10^{-1}	142	141	133	131
10^{-3}	262	260	251	250
10^{-5}	303	301	251	252
10^{-7}	331	330	252	253
10^{-9}	353	353	252	253

Table 4. (n=250, m=50)

ε	GR-SR1$_1$	GR-SR1$_2$	GR-PSB-TR	GR-SR1-TR
10^{-1}	17	17	17	17
10^{-3}	243	243	238	238
10^{-5}	257	257	251	251
10^{-7}	271	271	251	251
10^{-9}	279	279	252	252

Table 5. (n=50, m=250)

ε	GR-SR1$_1$	GR-SR1$_2$	GR-PSB-TR	GR-SR1-TR
10^{-1}	51	51	50	50
10^{-3}	66	66	51	51
10^{-5}	75	75	51	51
10^{-7}	82	82	52	52
10^{-9}	87	87	52	52

We make experiments with different parameters. We set these parameters range from small to large sizes to validate performance of our algorithm. As we can see from Tables 1-5, our algorithm (GR-SR1-TR) is competitive with the existing Greedy Quasi-Newton methods in terms of the number of iterations in achieving the same desired accuracy. Specifically, our algorithm maintains a clear advantage to previous Greedy SR1 methods and its performance is similar to the Greedy PSB method with Trust Region framework which has global convergence as well, meaning that our method achieves the purpose of reducing the effect of modification.

We can see that all Greedy Quasi-Newton methods have two stage of convergence as pointed in[22,23,27]. In the first stage, they convergent linearly, while in the second stage, they convergent superlinearly and almost as fast as the Newton method. Although Greedy Quasi-Newton methods perform slower in the beginning of iterations, with only linear convergence rate, these algorithms still maintain its competitiveness by decreasing the total number of iterations under the requirement of high accuracy with fast superlinear convergence in the local region.

Table 6. (n=m=50)

ε	GR-SR1$_1$	GR-SR1$_2$	GR-PSB-TR	GR-SR1-TR
10^{-1}	0.03s	0.03s	0.03s	0.03s
10^{-3}	0.04s	0.04s	0.05s	0.04s
10^{-5}	0.05s	0.05s	0.05s	0.05s
10^{-7}	0.06s	0.06s	0.07s	0.06s
10^{-9}	0.06s	0.06s	0.07s	0.06s

Table 7. (n=m=150)

ε	GR-SR1$_1$	GR-SR1$_2$	GR-PSB-TR	GR-SR1-TR
10^{-1}	0.73s	0.73s	0.85s	0.87s
10^{-3}	1.34s	1.33s	1.65s	1.43s
10^{-5}	1.44s	1.37s	1.65s	1.46s
10^{-7}	1.51s	1.43s	1.73s	1.74s
10^{-9}	1.61s	1.58s	1.73s	1.74s

Table 8. (n=m=250)

ε	GR-SR1$_1$	GR-SR1$_2$	GR-PSB-TR	GR-SR1-TR
10^{-1}	1.87s	1.83s	2.79s	1.95s
10^{-3}	3.44s	3.39s	5.62s	3.89s
10^{-5}	3.93s	3.91s	5.62s	4.51s
10^{-7}	4.35s	4.34s	5.82s	5.15s
10^{-9}	4.63s	4.57s	5.82s	5.15s

Table 9. (n=250, m=50)

ε	GR-SR1$_1$	GR-SR1$_2$	GR-PSB-TR	GR-SR1-TR
10^{-1}	0.14s	0.14s	0.21s	0.14s
10^{-3}	1.59s	1.59s	2.73s	1.63s
10^{-5}	1.65s	1.61s	3.51s	1.72s
10^{-7}	1.71s	1.65s	3.51s	1.72s
10^{-9}	1.85s	1.85s	3.75s	1.79s

Table 10. (n=50, m=250)

ε	GR-SR1$_1$	GR-SR1$_2$	GR-PSB-TR	GR-SR1-TR
10^{-1}	0.09s	0.09s	0.09s	0.09s
10^{-3}	0.13s	0.13s	0.12s	0.11s
10^{-5}	0.14s	0.14s	0.12s	0.11s
10^{-7}	0.16s	0.16s	0.13s	0.13s
10^{-9}	0.17s	0.16s	0.13s	0.13s

The variation about norm of the gradient with respect to time is shown in Tables 5-10 for different parameters, where *s* means seconds and we retain the last two decimal places. As it can be seen from the tables, the time to reach the same accuracy still shows only small gaps between our algorithm and the other algorithms in time cost.

In addition, during the experiments, it was found that the iteration process maintains the discriminant criterion (18) in most cases provided that the initial condition satisfies (18).

Besides, we need to point that for the condition $B \succeq G$, experiments show that it does not hold in most cases if take no correction technique. Therefore, the correction technique in [22,23] may be high-demanded, which leads to a significant reduction to the practical convergence rate. In the other hand, for some functions with strong self-concordant properties, the tight coefficient may not be computable directly, which makes the correction method relying on this specific coefficient lose the convergence guarantee and the explicit upper bound in these cases. In particular, this problem is more acute when the objective function is not a strong self-concordant function.

In contrast, our algorithm, GR-SR1-TR, which operates in the Trust Region framework, maintains convergence and provides explicit upper bounds even when the strong self-concordant coefficients cannot be computed directly. This property makes the algorithm equally robust and reliable when functions are not strong self-concordant. Thus, our algorithm has wider applicability and more effectiveness in practical applications.

5. Conclusion

In this paper, we prove that the approximation matrix sequence exhibits a linear descent property under the Frobenius-norm metric. Further, we integrate the Greedy technique with the Trust Region framework. We not only ensure the global convergence but also maintain an explicit convergence rate under specific conditions. We also improved the correction technique, deducing the influence of correction. To validate the effectiveness of our algorithm, we conducted a series of experiments. The experimental results show that our algorithm exhibits the expected performance in practical applications, which not only converges faster than the existing Greedy SR1 methods but also converges for general functions beyond strong self-concordant functions. These results confirm the effectiveness of our algorithm and provide new insights for researches in related fields.

Acknowledgements

This work was supported in part by my mentor, Liping Wang, and my senior brother, Hao Wu. I am very grateful for their help and guidance in my research process.

References

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

[2] R. Fletcher, M. J. D. Powell, A rapidly convergent descent method for minimization. Computer journal, 6(2), 163–168, 1963.

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

[4] R. Fletcher, A new approach to variable metric algorithms. Computer journal, 13(3), 317–322, 1970.

[5] D.F. Shanno, Conditioning of Quasi-Newton methods for function minimization. Mathematics of computation, 24, 647–656, 1970.

[6] A. R. Conn, N. I. M. Gould, P. L. Toint, Convergence of Quasi-Newton matrices generated by the symmetric rank one update. Mathematical Programming, 50(1–3), 177–195, 1991.

[7] H. L. Krauss, C. W. Bostian, and F. H. Raab, Solid State Radio Engineering, New York: J. Wiley & Sons, 1980.

[8] Y. H. Dai, Convergence properties of the BFGS algorithm. SIAM Journal on optimization, 13, 693–701, 2003.

[9] Y. H. Dai, A perfect example for the BFGS method. Mathematical Programming, 138, 501–530, 2013.

[10] D. H. Li, M. Fukushima, On the global convergence of the BFGS method for nonconvex unconstrained optimization problems. SIAM Journal on optimization, 11, 1054–1064, 2001.

[11] W. F. Mascarenhas, The BFGS method with exact line searches fails for non-convex objective functions. Mathematical Programming, 99, 49–61, 2004.

[12] J. J. Moré, J. A. Trangenstein, On the global convergence of Broyden's method. Mathematics of computation, 30, 523–540, 1976.

[13] C. G. Broyden, J. E. Dennis, J. J. Moré, On the local and superlinear convergence of Quasi-Newton methods. IMA Journal of Applied Mathematics, 12(3), 223–245, 1973.

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

[15] L. C. W. Dixon, Quasi-Newton algorithms generate identical points. Mathematical Programming, 2(1), 383–387, 1972.

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

[17] R. H. Byrd, D. C. Liu, J. Nocedal, On the behavior of Broyden's class of Quasi-Newton methods. SIAM Journal on optimization, 2(4), 533–557, 1992.

[18] J. Engels, H. Martínez, Local and superlinear convergence for partially known Quasi-Newton methods. SIAM Journal on optimization, 1(1), 42–56, 1991.

[19] W. Gao, D. Goldfarb, Quasi-Newton methods: superlinear convergence without line searches for self-concordant functions. Optimization Methods and Software, 34(1), 194–217, 2019.

[20] A. Mokhtari, M. Eisen, A. Ribeiro, IQN: an incremental quasi-Newton method with local superlinear convergence rate. SIAM Journal on optimization, 28(2), 1670–1698, 2018.

[21] H. Yabe, N. 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.

[22] A. Rodomanov, Y. Nesterov, Greedy Quasi-Newton methods with explicit superlinear convergence. SIAM Journal on optimization, 31(1), 785–811, 2021.

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

[24] A. Rodomanov, Y. Nesterov, Rates of superlinear convergence for classical Quasi-Newton methods. Mathematical Programming, 1–32, 2022.

[25] H. Ye, D. Lin, X. Chang, Z. Zhang, Towards explicit superlinear convergence rate for SR1. Mathematical Programming, 199(1): 1273–1303, 2023.

[26] Q. J. Jin, A. Mokhtari, Non-asymptotic superlinear convergence of standard quasi-Newton methods. Mathematical Programming, 200.1, 425–473, 2023.

[27] Z. Y. Ji, Y. H. Dai, Greedy PSB methods with explicit superlinear convergence. Computational Optimization and Applications, 85, 753 – 786, 2023.

[28] R. H. Byrd, H. F. Khalfan, R. B. Schnabel, Analysis of a symmetric rank-one trust region method. SIAM Journal on optimization, 6, 1025–1039, 1996.

[29] J. Nocedal, S. J. Wright, Numerical Optimization, New York, 1999.