OPT2020: 12th Annual Workshop on Optimization for Machine Learning

Incremental Greedy BFGS: An Incremental Quasi-Newton Method with Explicit Superlinear Rate

Zhan Gao GAOZHAN@SEAS.UPENN.EDU
Alec Koppel ALEC.E.KOPPEL.CIV@MAIL.MIL
Alejandro Ribeiro ARIBEIRO@SEAS.UPENN.EDU
University of Pennsylvania, Philadelphia, PA, USA
U.S. Army Research Laboratory, Adelphi, MD, USA

Abstract

Finite-sum minimization, i.e., problems where the objective may be written as the sum over a collection of instantaneous costs, are ubiquitous in modern machine learning and data science. Efficient numerical techniques for their solution must trade off per-step complexity with the number of steps required for convergence. Incremental Quasi-Newton methods (IQN) achieve a favorable balance of these competing attributes in the sense that their complexity is independent of the sample size, while their convergence rate can be faster than linear. This local superlinear behavior, to date, however, is known only asymptotically. In this work, we put forth a new variant of IQN, specifically of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) type, that incorporates a greedy basis vector selection step, and admits a non-asymptotic explicit local superlinear rate. To the best of our knowledge, this is the first time an explicit superlinear rate has been given for Quasi-Newton methods in the incremental setting.

1. Formulation and Context

Consider the finite-sum minimization problem, where the objective is the sum of a set of loss functions. That is, denote as $\mathbf{x} \in \mathbb{R}^p$ the decision variable in p-dimensions and $f_i(\mathbf{x}) : \mathbb{R}^p \to \mathbb{R}$ for $i = 1, \ldots, n$ as the constituent costs, which are assumed convex [19]. The goal is to compute the minimizer $\mathbf{x}^*$ of the cumulative function $f = \sum_i f_i$, i.e.,

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}} \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}) \ . \tag{1}$$

This problem subsumes numerous machine learning problems such as maximum likelihood and a posteriori estimation (MLE and MAP) [4], support vector machines [9], and various forms of unsupervised learning when given a fixed finite training data set [18]. Specifically, in empirical risk minimization for supervise learning, we have access to the training set $\{(\mathbf{z}_i, \mathbf{y}_i)\}_{i=1}^n$ and $f_i(\mathbf{x})$ represents the model fitness of $\mathbf{x}$ at $(\mathbf{z}_i, \mathbf{y}_i)$. The training loss is then the average performance over training samples.

In this work, we focus on instances of (1) when n may be large-scale, which makes computing the objective $f(\mathbf{x})$, the gradient $\nabla f(\mathbf{x})$, and the Hessian $\nabla^2 f(\mathbf{x})$ computationally intensive. In such cases, online or *incremental* algorithms are of interest which are able to operate on only *subsets* of functions [3] per step. In particular, a generic incremental method to solve (1) is one in which the

© Z. Gao, A. Koppel & A. Ribeiro.

update takes the form

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \alpha^t \mathbf{d}^t \ ,$$

where $\alpha^t > 0$ is a scalar step-size and $\mathbf{d}^t$ is a (approximate) descent direction on the aggregate cost $f(\mathbf{x})$ that is computed using only a mini-batch of $1 \leq B \ll n$ samples/functions. In incremental gradient methods, $\mathbf{d}^t$ is selected as the negative gradient direction associated with f_{i^t} where the index is selected uniformly at random from the training indices $i^t \sim \mathrm{U}\{1, \ldots, n\}$ or cyclically. Doing so, however, results in slow (sublinear) convergence [2] [13]. Efforts to ameliorate this issue by recursive averaging exist – one may either average the gradient [14, 16] or the decision-variable [22, 25] in order to achieve faster convergence, specifically, at up to a linear rate.

This behavior is far-surpassed by methods that incorporate second-order information, specifically Newton's method. Newton's method exhibits quadratic convergence in a region, although the $\mathcal{O}(n^2)$ computational effort required per step to compute the Hessian (second-derivative matrix) renders it inoperable when n is large. Quasi-Newton schemes in the batch setting approximate the Hessian inverse in Newton steps [7, 20], which reduces the per-step complexity to $\mathcal{O}(np^2)$, where p is the dimension of function variable, and can achieve a rate that is locally *superlinear* [7, 11].

Germaine to this paper specifically are efforts to alleviate the dependence on the sample size n entirely via incremental updates [15, 23]. That is, incremental Quasi-Newton schemes can achieve convergence that is locally superlinear, while having per-step complexity $\mathcal{O}(p^2)$, which is notably independent of sample size n. It is for this reason that interest in this family of methods has been spiking in recent years [1, 8, 17, 21, 27–29]. Their convergence behavior in the incremental setting, to date, however, is known only in an asymptotic sense, that is, to satisfy the Dennis-Moré condition [10], which is sufficient for local superlinear convergence [6]. This fact belies superior performance in practice. In this work, by incorporating a greedy basis vector selection step into incremental Quasi-Newton updates, we develop a method whose local superlinear rate may be characterized in an *explicit non-asymptotic* sense. This greedy step was developed in [24] for batch settings. Here we generalize it to the incremental setting, and is thus applicable to large-scale ERM (1). All proofs are deferred to the forthcoming journal version.

2. Incremental Greedy BFGS

We propose the **I**ncremental **G**reedy BFG**S** (IGS) method to address (1) for large n. The key aspects of that distinguish IGS from its non-incremental (batch) variant is that it is incremental and aggregated. It is incremental since it only updates the information of a single function selected at each iteration [2], and it is aggregated since the information of all functions is aggregated and used to update decision variable [14]. The former saves the cost per iteration, and the latter improves the variable update progress. We formally state the IGS in the following.

Initialization: Let $\mathbf{x}^0$ be the initial decision variable, $\mathbf{z}_1^0 = \ldots = \mathbf{z}_n^0 = \mathbf{x}_0$ be initial local variables associated with loss functions $\{f_i(\mathbf{x})\}_{i=1}^n$, and $\{\nabla f_i(\mathbf{z}_i^0)\}_{i=1}^n$ be gradients of $\{f_i(\mathbf{x})\}_{i=1}^n$ at $\{\mathbf{z}_i^0\}_{i=1}^n$. Let also $\{\mathbf{B}_i^0\}_{i=1}^n$ be initial Hessian approximations satisfying $\mathbf{B}_i^0 \succeq \nabla^2 f_i(\mathbf{z}_i^0)$ for $i = 1, \ldots, n$. Proceeding from this initialization, the IGS is divided into two steps: variable update and Hessian approximation update.

Variable update: At iteration t, denote as $\{\mathbf{z}_i^t, \nabla f_i(\mathbf{z}_i^t), \mathbf{B}_i^t\}_{i=1}^n$ the local variables, gradients and Hessian approximations of loss functions $\{f_i(\mathbf{x})\}_{i=1}^n$. We update the decision variable $\mathbf{x}^t$ by jointly

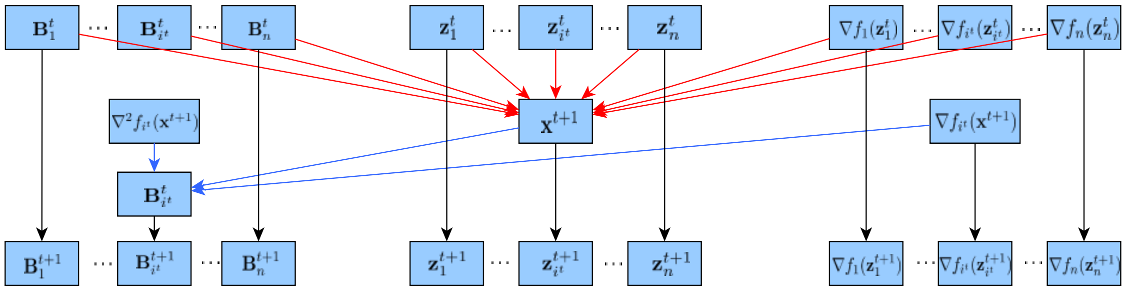

Figure 1: Incremental greedy BFGS (IGS): At iteration t with a selected index i^t, local variables, gradients and Hessian approximations of all functions are aggregated to update the variable $\mathbf{x}^{t+1}$ (red arrows). The Hessian approximation $\mathbf{B}_{i^t}^{t+1}$ is updated using the greedy BFGS (blue arrows). The terms $\mathbf{z}_{i^t}^{t+1}$ and $\nabla f_{i^t}(\mathbf{z}_{i^t}^{t+1})$ are updated as $\mathbf{x}^{t+1}$ and $\nabla f_{i^t}(\mathbf{x}^{t+1})$, whereas all other $\mathbf{z}_j^{t+1}$ and $\nabla f_j(\mathbf{z}_j^{t+1})$ are untouched (black arrows).

using such information. In particular, the second order approximation of $f_i(\mathbf{x})$ at $\mathbf{z}_i^t$ is given by

$$f_i(\mathbf{x}) = f_i(\mathbf{z}_i^t) + \nabla f_i(\mathbf{z}_i^t)^\top(\mathbf{x} - \mathbf{z}_i^t) + \frac{1}{2}(\mathbf{x} - \mathbf{z}_i^t)^\top \nabla^2 f_i(\mathbf{z}_i^t)(\mathbf{x} - \mathbf{z}_i^t). \tag{2}$$

By approximating the Hessian $\nabla^2 f_i(\mathbf{z}_i^t)$ with $\mathbf{B}_i^t$ and aggregating all loss functions $\{f_i\}_{i=1}^t$, the objective function $f(\mathbf{x})$ is approximated with a second-order Taylor's expansion

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n \left[f_i(\mathbf{z}_i^t) + \nabla f_i(\mathbf{z}_i^t)^\top(\mathbf{x} - \mathbf{z}_i^t) + \frac{1}{2}(\mathbf{x} - \mathbf{z}_i^t)^\top \nabla^2 f_i(\mathbf{z}_i^t)(\mathbf{x} - \mathbf{z}_i^t)\right]. \tag{3}$$

We define the updated variable $\mathbf{x}^{t+1}$ as the minimizer of the quadratic function (3)

$$\mathbf{x}^{t+1} = \left(\frac{1}{n}\sum_{i=1}^n \mathbf{B}_i^t\right)^{-1}\left[\frac{1}{n}\sum_{i=1}^n \mathbf{B}_i^t \mathbf{z}_i^t - \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{z}_i^t)\right] \tag{4}$$

where local variables, gradients and Hessian approximations of all functions are used for the variable update, in order to reduce the stochastic approximation error. While the update in (4) looks computationally prohibitive, it can be implemented with complexity independent of n, as we detail later in this section. We then use $\mathbf{x}^{t+1}$ to update local variables $\{\mathbf{z}_i^t\}$. Let i^t be the selected function index at iteration t in a cyclic scheme. We only update the information of this function while keeping the others simply unchanged

$$\mathbf{z}_{i^t}^{t+1} = \mathbf{x}^{t+1}, \ \mathbf{z}_i^{t+1} = \mathbf{z}_i^t \text{ for all } i \neq i^t. \tag{5}$$

Hessian approximation update: We continue to update Hessian approximations following the same scheme as local variables [cf. (5)]. In other words, the Hessian approximation $\mathbf{B}_{i^t}^t$ of the selected function $f_{i^t}(\mathbf{x})$ is updated with the greedy BFGS, while the others are kept as their previous values. To do so, we first define $d^t = \|\mathbf{z}_{i^t}^{t+1} - \mathbf{z}_{i^t}^t\|_{\nabla^2 f_{i^t}(\mathbf{z}_{i^t}^t)}$ and compute $\hat{\mathbf{B}}_{i^t}^t = (1 + C_M d^t)\mathbf{B}_{i^t}^t \succeq \nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1})$ to well define the greedy BFGS update [24], where $\|\cdot\|_{\nabla^2 f_{i^t}(\mathbf{z}_{i^t}^t)}$ is the operator norm

Algorithm 1: Incremental Greedy BFGS (IGS) Method

```
Input: Loss functions $\{f_i(\mathbf{x})\}_{i=1}^n$, initial decision vector $\mathbf{x}_0$, initial Hessian approximations
        $\{\mathbf{B}_i^0\}_{i=1}^n$, and initial local variables $\mathbf{z}_i^0 = \mathbf{x}^0$ for $i = 1, \dots, n$
for $t = 0, 1, \dots, T$ do
    Compute gradients $\nabla f_i(\mathbf{z}_i^t)$ for $i = 1, \dots, n$ and update the decision variable
    $\mathbf{x}^{t+1} = \big(\frac{1}{n}\sum_{i=1}^n \mathbf{B}_i^t\big)^{-1}\big[\frac{1}{n}\sum_{i=1}^n \mathbf{B}_i^t \mathbf{z}_i^t - \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{z}_i^t)\big]$;
    Select the index $i^t$ and update
    $\mathbf{z}_{i^t}^{t+1} = \mathbf{x}^{t+1}$, $\mathbf{z}_i^{t+1} = \mathbf{z}_i^t$ for all $i \neq i^t$;
    Compute $d^t = \|\mathbf{z}_{i^t}^{t+1} - \mathbf{z}_{i^t}^t\|_{\mathbf{z}_{i^t}^t}$, $\hat{\mathbf{B}}_{i_t}^t = (1 + C_M d^t)\mathbf{B}_{i_t}^t$ and $\nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1})$;
    Select the greedy variable variation
    $\mathbf{u}^t = \text{argmax}_{\mathbf{u} \in \{\mathbf{e}_1, \dots, \mathbf{e}_n\}} \frac{<\hat{\mathbf{B}}_{i_t}^t \mathbf{u}, \mathbf{u}>}{<\nabla^2 f_{i_t}(\mathbf{z}_{i^t}^{t+1})\mathbf{u}, \mathbf{u}>}$;
    Update the Hessian matrix [cf. (7)]
    $\mathbf{B}_{i^t}^{t+1} = \text{BFGS}\left(\hat{\mathbf{B}}_{i^t}^t, \mathbf{u}^t, \nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1})\mathbf{u}^t\right)$, $\mathbf{B}_i^{t+1} = \mathbf{B}_i^t$ for all $i \neq i^t$ ;
end
return $x^T$;
```

and C_M is strongly self-concordant constant [cf. (14) in Lemma 1]. We then select the variable variation $\mathbf{u}^t$ greedily

$$\mathbf{u}^t = \underset{\mathbf{u} \in \{\mathbf{e}_1, \dots, \mathbf{e}_n\}}{\text{argmax}} \frac{< \hat{\mathbf{B}}_{i^t}^t \mathbf{u}, \mathbf{u} >}{< \nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1})\mathbf{u}, \mathbf{u} >}, \tag{6}$$

and compute the corresponding gradient variation $\mathbf{y}^t = \nabla^2 f(\mathbf{z}_{i^t}^{t+1})\mathbf{u}^t$. This is the key departure from existing incremental Quasi-Newton schemes which do not employ basis vector selections. Here $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ denote the coordinate orthogonal basis, that is, $\mathbf{e}_k = [0; \cdots 0; 1; 0 \cdots 0] \in \mathbb{R}^n$, with the k-th entry as the only non-zero. By substituting greedy $\mathbf{u}^t$ and $\mathbf{y}^t$ into the BFGS update originally defined in [5, 12, 26], we update the Hessian estimate as

$$\mathbf{B}_{i^t}^{t+1} = \hat{\mathbf{B}}_{i^t}^t - \frac{\hat{\mathbf{B}}_{i^t}^t \mathbf{u}^t (\mathbf{u}^t)^\top \hat{\mathbf{B}}_{i^t}^t}{(\mathbf{u}^t)^\top \hat{\mathbf{B}}_{i^t}^t \mathbf{u}^t} + \frac{\nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1})\mathbf{u}^t (\mathbf{u}^t)^\top \nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1})}{(\mathbf{u}^t)^\top \nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1})\mathbf{u}^t}, \ \mathbf{B}_i^{t+1} = \mathbf{B}_i^t \text{ for all } i \neq i^t. \tag{7}$$

Thus far we have completed updating $\mathbf{z}_i^{t+1}$ and $\mathbf{B}_i^{t+1}$, and iteration $t+1$ follows similarly. Figure 1 shows the processing architecture defined by IGS, which is formally summarized as Algorithm 1.

Efficient implementation: The IGS can be implemented in an efficient manner. In particular, the IGS requires the computation of $(\sum_{i=1}^n \mathbf{B}_i^t)^{-1}$, $\sum_{i=1}^n \mathbf{B}_i^t \mathbf{z}_i^t$ and $\sum_{i=1}^n \nabla f_i(\mathbf{z}_i^t)$ to perform the update (4). Suppose that at iteration t, only the information of function $f_{i^t}(\mathbf{x})$ is updated such that the latter two variables can be evaluated for iteration $t+1$ as

$$\sum_{i=1}^n \mathbf{B}_i^{t+1} \mathbf{z}_i^{t+1} = \sum_{i=1}^n \mathbf{B}_i^t \mathbf{z}_i^t + \mathbf{B}_{i^t}^{t+1} \mathbf{z}_{i^t}^{t+1} - \mathbf{B}_{i^t}^t \mathbf{z}_{i^t}^t, \ \sum_{i=1}^n \nabla f_i(\mathbf{z}_i^t) = \sum_{i=1}^n \nabla f_i(\mathbf{z}_i^t) + \nabla f_{i^t}(\mathbf{z}_{i^t}^{t+1}) - \nabla f_{i^t}(\mathbf{z}_{i^t}^t) \tag{8}$$

such that only $\mathbf{B}_{i^t}^{t+1}$ and $\nabla f_{i^t}(\mathbf{z}_{i^t}^{t+1})$ are required for computation corresponding to the cost on the order of $\mathcal{O}(p^2)$ and $\mathcal{O}(p)$, respectively. With respect to evaluating $(\sum_{i=1}^n \mathbf{B}_i^{t+1})^{-1}$, we first update

$\sum_{i=1}^{n} \mathbf{B}_i^{t+1}$ similarly as (8)

$$\sum_{i=1}^{n} \mathbf{B}_i^{t+1} = \sum_{i=1}^{n} \mathbf{B}_i^{t} + \mathbf{B}_{i^t}^{t+1} - \mathbf{B}_{i^t}^{t} = \sum_{i=1}^{n} \mathbf{B}_i^{t} - \frac{\hat{\mathbf{B}}_{i^t}^{t} \mathbf{u}^t (\mathbf{u}^t)^\top \hat{\mathbf{B}}_{i^t}^{t}}{(\mathbf{u}^t)^\top \hat{\mathbf{B}}_{i^t}^{t} \mathbf{u}^t} + \frac{\nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1}) \mathbf{u}^t (\mathbf{u}^t)^\top \nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1})}{(\mathbf{u}^t)^\top \nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1}) \mathbf{u}^t}. \tag{9}$$

With $\big(\sum_{i=1}^{n} \mathbf{B}_i^t\big)^{-1}$ given at iteration t, we can compute $\big(\sum_{i=1}^{n} \mathbf{B}_i^{t+1}\big)^{-1}$ by applying the Sherman-Morrison formula twice to (9) as

$$\Big(\sum_{i=1}^{n} \mathbf{B}_i^{t+1}\Big)^{-1} = \mathbf{S}^t + \frac{\mathbf{S}^t (\hat{\mathbf{B}}_{i^t}^{t} \mathbf{u}^t)(\hat{\mathbf{B}}_{i^t}^{t} \mathbf{u}^t)^\top \mathbf{S}^t}{(\mathbf{u}^t)^\top \hat{\mathbf{B}}_{i^t}^{t} \mathbf{u}^t - (\hat{\mathbf{B}}_{i^t}^{t} \mathbf{u}^t)^\top \mathbf{S}^t (\hat{\mathbf{B}}_{i^t}^{t} \mathbf{u}^t)} \tag{10}$$

with

$$\mathbf{S}^t = \Big(\sum_{i=1}^{n} \mathbf{B}_i^{t}\Big)^{-1} - \frac{\big(\sum_{i=1}^{n} \mathbf{B}_i^{t}\big)^{-1} \nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1}) \mathbf{u}^t (\mathbf{u}^t)^\top \nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1}) \big(\sum_{i=1}^{n} \mathbf{B}_i^{t}\big)^{-1}}{(\mathbf{u}^t)^\top \nabla^2 f_{i^t}(\mathbf{z}_{i^t}^{t+1}) \mathbf{u}^t + (\mathbf{u}^t)^\top \nabla^2 f(\mathbf{z}_{i^t}^{t+1}) \big(\sum_{i=1}^{n} \mathbf{B}_i^{t}\big)^{-1} \nabla^2 f(\mathbf{z}_{i^t}^{t+1}) \mathbf{u}^t}. \tag{11}$$

Here, (10) and (11) use the preliminary knowledge $\big(\sum_{i=1}^{n} \mathbf{B}_i^t\big)^{-1}$ to update $\big(\sum_{i=1}^{n} \mathbf{B}_i^{t+1}\big)^{-1}$, which avoids computing the matrix inverse and results in the computation cost $\mathcal{O}(p^2)$. Together with (8) and (10), the overall cost of IGS is then $\mathcal{O}(p^2)$, which is substantially reduced compared to its $\mathcal{O}(np^2)$ batch counterpart. Next we shift gears to presenting our convergence results.

3. Convegence Analysis

We shift to presenting our main contribution: the explicit local superlinear convergence of the IGS. To develop these results, some conditions on the functions $\{f_i(\mathbf{x})\}_{i=1}^n$ are required, as stated next.

Assumption 1 *Consider the loss functions $\{f_i(\mathbf{x})\}_{i=1}^n$ in (1). There exist positive constants $0 < \mu < L$ such that, for all $i = 1, \ldots, n$ and any $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^p$, it holds that*

$$\mu \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \leq (\nabla f_i(\mathbf{x}) - \nabla f_i(\hat{\mathbf{x}}))^\top (\mathbf{x} - \hat{\mathbf{x}}) \leq L \|\mathbf{x} - \hat{\mathbf{x}}\|^2. \tag{12}$$

Assumption 2 *Consider the loss functions $\{f_i(\mathbf{x})\}_{i=1}^n$ in (1). There exists a positive constant $C_L > 0$ such that, for all $i = 1, \ldots, n$ and any $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^p$, it holds that*

$$\|\nabla^2 f_i(\mathbf{x}) - \nabla^2 f_i(\hat{\mathbf{x}})\| \leq C_L \|\mathbf{x} - \hat{\mathbf{x}}\|. \tag{13}$$

Assumption 1 indicates that each loss function $f_i(\mathbf{x})$ is strongly convex with respect to μ and its gradient $\nabla f_i(\mathbf{x})$ is Lipschitz continuous with respect to L. Assumption 2 implies that the Hessian $\nabla^2 f_i(\mathbf{x})$ is Lipschitz continuous with respect to C_L. Furthermore, with Assumptions 1-2, we can refer that the functions $\{f_i(\mathbf{x})\}_{i=1}^n$ are strongly self-concordant formally stated as follows.

Lemma 1 *Consider the loss function $f(\mathbf{x})$ satisfying Assumptions 1-2. Then $f(\mathbf{x})$ is strongly self-concordant, i.e., there exists a constant $C_M > 0$ such that for any $\mathbf{x}, \hat{\mathbf{x}}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^p$, it holds that*

$$\nabla^2 f(\mathbf{x}) - \nabla^2 f(\hat{\mathbf{x}}) \preceq C_M \|\mathbf{x} - \hat{\mathbf{x}}\|_{\nabla^2 f(\mathbf{y})} \nabla^2 f(\mathbf{z}). \tag{14}$$

where $\|\cdot\|_{\nabla^2 f(\mathbf{y})}$ is the the operator norm.

Lemma 1 states that a strongly convex function with Lipschitz Hessian is strongly self-concordant, ensuring losses evaluated at distinct variables $\mathbf{x}$ and $\hat{\mathbf{x}}$ are smooth.

With these preliminaries addressed, we shift to discussing the convergence. We do so in terms of the standard criterion $\|\mathbf{x}^t - \mathbf{x}^*\|$, where $\mathbf{x}^*$ is the optimal solution of (1). Our goal is to show the error sequence $\|\mathbf{x}^t - \mathbf{x}^*\|$ generated by the IGS converges to zero at an explicit superlinear rate. We start by showing a linear convergence rate in the following theorem, based on which we proceed to prove the non-asymptotic superlinear convergence.

Theorem 2 *Consider the IGS method. If Assumptions 1-2 hold, then for any $r \in (0,1)$, there exist positive constants $\epsilon(r) > 0$ and $\sigma(r) > 0$ such that if the initialization satisfies $\|\mathbf{x}^0 - \mathbf{x}^*\| \leq \epsilon(r)$ and $\operatorname{tr}\Big(\nabla^2 f_i(\mathbf{z}_i^0)^{-1}\big(\mathbf{B}_i^0 - \nabla^2 f_i(\mathbf{z}_i^0)\big)\Big) \leq \sigma(r)$ for all $i = 1, \ldots, n$, the error sequence satisfies*

$$\|\mathbf{x}^t - \mathbf{x}^*\| \leq r^{\lfloor \frac{t-1}{n} \rfloor + 1} \|\mathbf{x}^0 - \mathbf{x}^*\| \tag{15}$$

where $\operatorname{tr}\Big(\nabla^2 f_i(\mathbf{z}_i^0)^{-1}\big(\mathbf{B}_i^0 - \nabla^2 f_i(\mathbf{z}_i^0)\big)\Big)$ is the sum of eigenvalues of the Hessian approximation error $\mathbf{B}_i^0 - \nabla^2 f_i(\mathbf{z}_i^0)$ with respect to the inverse Hessian $\nabla^2 f_i(\mathbf{z}_i^0)^{-1}$ and $\lfloor \cdot \rfloor$ is the floor function.

Theorem 2 shows that the error sequence $\|\mathbf{x}^t - \mathbf{x}^*\|$ generated by the IGS converges at a linear rate after each pass over all functions. This is a local linear convergence, i.e., the conditions $\|\mathbf{x}^0 - \mathbf{x}^*\| \leq \epsilon(r)$ and $\operatorname{tr}\big(\nabla^2 f_i(\mathbf{z}_i^0)^{-1}\big(\mathbf{B}_i^0 - \nabla^2 f_i(\mathbf{z}_i^0)\big)\big) \leq \sigma(r)$ assume that the initialization is close to the optimal. We employ induction to establish the explicit superlinear rate.

Theorem 3 *Consider the same settings as Theorem 2. Let r be the linear rate in (15), D be the constant depending on loss function properties, and k_0 be such that $(1 - \frac{\mu}{pL})^{k_0} D \leq 1$. Then, the error sequence satisfies*

$$\|\mathbf{x}^t - \mathbf{x}^*\| \leq (1 - \frac{\mu}{pL})^{\frac{k(k+1)}{2}} r^{k_0+1} \|\mathbf{x}^0 - \mathbf{x}^*\| \tag{16}$$

where $k = \lfloor \frac{t-1}{n} \rfloor - k_0$ with $\lfloor \cdot \rfloor$ the floor function.

Theorem 3 establishes that the sequence of variables of the IGS converges to the optimal solution at an explicit superlinear rate after each pass over all functions. Consequently, we obtain that subsequences $\{\|\mathbf{z}_i^{kn+i} - \mathbf{x}^*\|\}_{k=0}^{\infty}$ for $i = 1, \ldots, n$ converge to zero at explicit superlinear rates. This extends the results in [24], i.e., the explicit superlinear rate of greedy Quasi-Newton methods, to the incremental (stochastic) setting in large-scale optimization problems, and establishes that the stochasticity does not harm its non-asymptotic superlinear convergence nature. It further contrasts the asymptotic-only incremental superlinear rates presented in [15, 23].

References

[1] Albert S Berahas, Majid Jahani, and Martin Takáč. Quasi-newton methods for deep learning: Forget the past, just sample. *arXiv preprint arXiv:1901.09997*, 2019.

[2] Dimitri P Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011.

[3] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pages 161–168, 2008.

[4] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[5] 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.

[6] 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.

[7] 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.

[8] Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

[9] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

[10] 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.

[11] Wenbo Gao and Donald Goldfarb. Quasi-newton methods: superlinear convergence without line searches for self-concordant functions. *Optimization Methods and Software*, 34(1):194–217, 2019.

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

[13] Mert Gürbüzbalaban, Asu Ozdaglar, and Pablo Parrilo. Convergence rate of incremental gradient and newton methods. *arXiv preprint arXiv:1510.08562*, 2015.

[14] Mert Gurbuzbalaban, Asuman Ozdaglar, and Pablo A Parrilo. On the convergence rate of incremental aggregated gradient algorithms. *SIAM Journal on Optimization*, 27(2):1035–1048, 2017.

[15] Aryan Mokhtari, Mark Eisen, and Alejandro Ribeiro. Iqn: An incremental quasi-newton method with local superlinear convergence rate. *SIAM Journal on Optimization*, 28(2):1670–1698, 2018.

[16] Aryan Mokhtari, Mert Gurbuzbalaban, and Alejandro Ribeiro. Surpassing gradient descent provably: A cyclic incremental method with linear convergence rate. *SIAM Journal on Optimization*, 28(2):1420–1447, 2018.

[17] Philipp Moritz, Robert Nishihara, and Michael Jordan. A linearly-convergent stochastic l-bfgs algorithm. In *Artificial Intelligence and Statistics*, pages 249–258, 2016.

[18] Kevin P Murphy. *Machine learning: a probabilistic perspective*. 2012.

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

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

[21] Mert Pilanci and Martin J Wainwright. Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization*, 27(1):205–245, 2017.

[22] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

[23] Anton Rodomanov and Dmitry Kropotov. A superlinearly-convergent proximal newton-type method for the optimization of finite sums. In *International Conference on Machine Learning*, pages 2597–2605, 2016.

[24] Anton Rodomanov and Yurii Nesterov. Greedy quasi-newton methods with explicit superlinear convergence. *arXiv preprint arXiv:2002.00657*, 2020.

[25] David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.

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

[27] Hoi-To Wai, Wei Shi, César A Uribe, Angelia Nedić, and Anna Scaglione. Accelerating incremental gradient optimization with curvature information. *Computational Optimization and Applications*, pages 1–34, 2020.

[28] Farzad Yousefian, Angelia Nedi?, and Uday V Shanbhag. On stochastic and deterministic quasi-newton methods for nonstrongly convex optimization: Asymptotic convergence and rate analysis. *SIAM Journal on Optimization*, 30(2):1144–1172, 2020.

[29] Chaoxu Zhou, Wenbo Gao, and Donald Goldfarb. Stochastic adaptive quasi-newton methods for minimizing expected values. In *International Conference on Machine Learning*, pages 4150–4159, 2017.