AN INCREMENTAL QUASI-NEWTON METHOD WITH A LOCAL SUPERLINEAR CONVERGENCE RATE

Aryan Mokhtari *Mark Eisen* *Alejandro Ribeiro*

Department of Electrical and Systems Engineering, University of Pennsylvania

ABSTRACT

We present an incremental Broyden-Fletcher-Goldfarb-Shanno (BFGS) method as a quasi-Newton algorithm with a cyclically iterative update scheme for solving large-scale optimization problems. The proposed incremental quasi-Newton (IQN) algorithm reduces computational cost relative to traditional quasi-Newton methods by restricting the update to a single function per iteration and relative to incremental second-order methods by removing the need to compute the inverse of the Hessian. A local superlinear convergence rate is established and a strong improvement is shown over first order methods numerically for a set of common large-scale optimization problems.

Index Terms— Stochastic optimization, quasi-Newton, incremental method, superlinear convergence

1. INTRODUCTION

We study a large scale optimization problem where we seek to minimize an objective function that is an aggregation of many component objective functions. To be more precise, consider a variable $\mathbf{x} \in \mathbb{R}^p$ and a function f which is defined as the average of n strongly convex functions labelled $f_i : \mathbb{R}^p \to \mathbb{R}$ for $i = 1, \ldots, n$. Our goal is to find the optimal point $\mathbf{x}^*$ as the solution to the problem

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}), \tag{1}$$

in a computationally efficient manner even when n is large. Problems of this form arise in machine learning [1–4], control problems [5–7], and wireless communication [8–10].

Much of the theory that has been developed to solve (1) is centered on the use of iterative descent methods. For large scale machine learning and optimization settings, n can be very large and the full gradient is often too costly to compute at each iteration. As an alternative, stochastic methods seek to find a solution to (1) while computing only a subset of the total n gradients at each iteration, thus significantly reducing the computational burden. The simplest version of a stochastic algorithm is stochastic gradient descent (SGD), which at each time step t draws a random index i_t and performs a descent using the gradient of only a single function [2]. More sophisticated stochastic first order methods use variance-reduced gradients (e.g. SVRG [11]) or averaging gradients (e.g. SAG [12, 13], SAGA [14]) to reduce computation cost while retaining a linear convergence rate. Moving beyond first order information, there have been stochastic quasi-Newton methods to approximate Hessian information [15–19]. All of these stochastic quasi-Newton methods reduce computational cost of quasi-Newton methods by updating only a randomly chosen single or small subset of gradients at each iteration. However, they are not able to recover the superlinear convergence rate of quasi-Newton methods [20–22].

Alternatively to stochastic methods are incremental methods, which deterministically iterate through all component functions in a cyclic fashion, again only computing a single gradient or Hessian per iteration. Incremental methods have been used with both aggregated gradients [23, 24] and second-order Hessian information [25, 26] . The incremental Newton method (NIM) in [26] is the only incremental method shown to have a superlinear convergence rate; however, the Hessian function is not always available or computationally feasible. Moreover, the implementation of NIM requires computation of the incremental aggregated Hessian inverse which has the computational complexity of the order $\mathcal{O}(p^3)$.

We propose an incremental variant of BFGS which achieves a greater local convergence rate than that of first order incremental or stochastic methods while reducing the computational cost per iteration to $\mathcal{O}(p^2)$ by using an incremental aggregated approximation of the Hessian.

We begin with the paper by introducing the well-known BFGS method and its update for solving (1) (Section 2). While incremental methods have been used in first order methods, they achieve only a linear convergence rate. We introduce an incremental quasi-Newton (IQN) method that requires only computing a single gradient and Hessian approximation per iteration (Section 3). The proposed IQN method uses an approximation of second-order information that requires less cost than computing the Hessian inverse directly while maintaining its superior analytical and numerical performance. The local superlinear convergence of IQN is established (Section 4) and performance advantages relative to first order stochastic and incremental methods are evaluated numerically (Section 5). Proofs for results presented are found in [27].

2. BFGS QUASI-NEWTON METHOD

Consider the problem in (1) for a relatively large n. In a conventional optimization setting, it can be solved using a descent method, such as gradient descent or Newton's method, which iteaitivley updates a variable $\mathbf{x}^t$ for $t = 0, 1, \ldots$ using the general recursive expression

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \eta^t \mathbf{d}^t, \tag{2}$$

where η^t is a scalar step size and $\mathbf{d}^t$ is the descent direction at time t—defined as either $\mathbf{d}^t := -\nabla f(\mathbf{x}^t)$ or $\mathbf{d}^t := -(\nabla^2 f(\mathbf{x}^t))^{-1}\nabla f(\mathbf{x}^t)$ for gradient descent and Newton's method, respectively. In both cases, iterations $\mathbf{x}^t$ converge to the optimal solution $\mathbf{x}^*$. Gradient descent, however, using only the first-order information contained in the gradient converges at a linear rate while Newton's method, using second-order information in the Hessian, converges at a significantly faster quadratic rate [28].

In the case that the Hessian information required in Newton's method is either unavailable or too costly to evaluate, quasi-Newton methods have been developed to approximate second-order information to improve upon the convergence rate of first order methods [20]. Quasi-newton methods perform an update using a descent direction of the form $\mathbf{d}^t := -(\mathbf{B}^t)^{-1}\nabla f(\mathbf{x}^t)$, where $\mathbf{B}^t$ is an approximation of the Hessian $\nabla^2 f(\mathbf{x}^t)$. There are different approaches to approximate the Hessian, with the most popular being the method of Broyden-Fletcher-Goldfarb-Shanno (BFGS) [20–22]. In BFGS, two auxiliary variables are defined to capture difference in variables and gradients of successive iterations, i.e.

$$\mathbf{s}^t := \mathbf{x}^{t+1} - \mathbf{x}^t, \qquad \mathbf{y}^t := \nabla f(\mathbf{x}^{t+1}) - \nabla f(\mathbf{x}^t). \tag{3}$$

Supported by NSF CAREER CCF-0952867 and ONR N00014-12-1-0997.

Then, given the variable variation $\mathbf{s}^t$ and gradient variation $\mathbf{y}^t$, the Hessian approximation is updated using the recursive equation

$$\mathbf{B}^{t+1} = \mathbf{B}^t + \frac{\mathbf{y}^t \mathbf{y}^{tT}}{\mathbf{y}^{tT}\mathbf{s}^t} - \frac{\mathbf{B}^t \mathbf{s}^t \mathbf{s}^{tT} \mathbf{B}^t}{\mathbf{s}^{tT}\mathbf{B}^t\mathbf{s}^t}. \tag{4}$$

The BFGS method is popular not only for its strong numerical performance relative to the gradient descent method, but also because it is shown to exhibit a superlinear convergence rate [20], thereby providing a theoretical guarantee of superior performance. In fact, it can be shown that, for the BFGS update the equation

$$\lim_{t\to\infty} \frac{\|(\mathbf{B}^t - \nabla^2 f(\mathbf{x}^*))\mathbf{s}^t\|}{\|\mathbf{s}^t\|} = 0 \tag{5}$$

known as the Dennis-Moré condition, which is both necessary and sufficient for superlinear convergence [22], is satisified. This result solidifies quasi-Newton methods as a strong alternative to first order methods when exact second-order information is unavailable.

3. INCREMENTAL QUASI-NEWTON METHOD

We introduce a novel incremental quasi-Newton (IQN) algorithm, in which the updated gradient information of only a single function f_i is updated at each iteration. Consider that each component Hessian $\nabla^2 f_i(\mathbf{x}^t)$ is approximated with $\mathbf{B}_i^t$ and define n copies of the variable $\mathbf{x}^t$, notated as $\mathbf{z}_1^t, \mathbf{z}_2^t, \ldots, \mathbf{z}_n^t$, each corresponding to a different function f_i. In particular, $\mathbf{z}_i^t$ shows the vector $\mathbf{x}$ at the last time before step t that function f_i is chosen to update descent direction.

We define the update of Hessian approximation $\mathbf{B}_i^t$ using the traditional BFGS update in (3)-(4), but instead using the i-th copies $\mathbf{z}_i^t$ and $\mathbf{z}_i^{t+1}$ in place of $\mathbf{x}^t$ and $\mathbf{x}^{t+1}$. We redefine the variable and gradient differences associated with the function f_i as

$$\mathbf{s}_i^t := \mathbf{z}_i^{t+1} - \mathbf{z}_i^t \qquad \mathbf{y}_i^t := \nabla f_i(\mathbf{z}_i^{t+1}) - \nabla f_i(\mathbf{z}_i^t), \tag{6}$$

Note that if at step $t+1$ and $t-\tau$, where $\tau \geq 0$, the function f_i is chosen for updating the gradient information, then we have $\mathbf{z}_i^{t+1} = \mathbf{x}^{t+1}$ and $\mathbf{z}_i^t = \mathbf{x}^{t-\tau}$. In particular, if we use a cyclic scheme for choosing the functions we have $\tau = n-1$, i.e., $\mathbf{z}_i^t = \mathbf{x}^{t-n+1}$. This observation shows that the variable and gradient variation defined in (6) can be written as $\mathbf{s}_i^t := \mathbf{x}^{t+1} - \mathbf{x}^{t-\tau}$ and $\mathbf{y}_i^t := \nabla f_i(\mathbf{x}_i^{t+1}) - \nabla f_i(\mathbf{x}_i^{t-\tau})$. Therefore, these definitions are different from the ones for classic BFGS in (3)-(4). Likewise, the Hessian approximation of the IQN is different from BFGS in using $\mathbf{s}_i^t$ and $\mathbf{y}_i^t$ instead of $\mathbf{s}^t$ and $\mathbf{y}^t$. Thus, the Hessian approximation $\mathbf{B}_i^t$ corresponding to the function f_i is updated as

$$\mathbf{B}_i^{t+1} = \mathbf{B}_i^t + \frac{\mathbf{y}_i^t \mathbf{y}_i^{tT}}{\mathbf{y}_i^{tT}\mathbf{s}_i^t} - \frac{\mathbf{B}_i^t \mathbf{s}_i^t \mathbf{s}_i^{tT} \mathbf{B}_i^t}{\mathbf{s}_i^{tT}\mathbf{B}_i^t\mathbf{s}_i^t}. \tag{7}$$

To derive the full variable update, we use an approach similar to that used in the Newton-type Incremental Method (NIM) [26]. Consider the second-order approximation of the function $f_i(\mathbf{x})$ centered around $\mathbf{z}_i^t$ as

$$\begin{aligned} f_i(\mathbf{x}) \approx\ & f_i(\mathbf{z}_i^t) + \nabla f_i(\mathbf{z}_i^t)^T(\mathbf{x} - \mathbf{z}_i^t) \\ & + \frac{1}{2}(\mathbf{x} - \mathbf{z}_i^t)^T \nabla^2 f_i(\mathbf{z}_i^t)(\mathbf{x} - \mathbf{z}_i^t). \end{aligned} \tag{8}$$

As in traditional quasi-Newton methods, we replace the i-th Hessian $\nabla^2 f_i(\mathbf{z}_i^t)$ by $\mathbf{B}_i^t$. Using the approximation matrices in place of Hessians, the objective function $f(\mathbf{x}) = (1/n)\sum_{i=1}^n f_i(\mathbf{x})$ can be approximated with

$$\begin{aligned} f(\mathbf{x}) \approx\ & \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{z}_i^t) + \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{z}_i^t)^T(\mathbf{x} - \mathbf{z}_i^t) \\ & + \frac{1}{n}\sum_{i=1}^n \frac{1}{2}(\mathbf{x} - \mathbf{z}_i^t)^T \mathbf{B}_i^t (\mathbf{x} - \mathbf{z}_i^t). \end{aligned} \tag{9}$$

Considering the approximation in (9), we approximate the optimal argument of $f(\mathbf{x})$ by the optimal argument of the function in (9). Thus, we define $\hat{\mathbf{x}}^{t+1} := \operatorname{argmin}\{(1/n)\sum_{i=1}^n f_i(\mathbf{z}_i^t) + (1/n)\sum_{i=1}^n \nabla f_i(\mathbf{z}_i^t)^T(\mathbf{x} - \mathbf{z}_i^t) + (1/2n)\sum_{i=1}^n (\mathbf{x} - \mathbf{z}_i^t)^T \mathbf{B}_i^t (\mathbf{x} - \mathbf{z}_i^t)\}$ as the auxialiary variable at iteration $t+1$. Solving this quadratic programming yields the following closed-form update for the variable $\hat{\mathbf{x}}^{t+1}$

$$\hat{\mathbf{x}}^{t+1} = \left(\frac{1}{n}\sum_{i=1}^n \mathbf{B}_i^t\right)^{-1} \left[\frac{1}{n}\sum_{i=1}^n \mathbf{B}_i^t \mathbf{z}_i^t - \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{z}_i^t)\right]. \tag{10}$$

As in traditional descent methods, we introduce a step size $0 \leq \eta^t \leq 1$ and define the variable $\mathbf{x}^{t+1}$ corresponding to step $t+1$ as the weighted average of the previous iterate $\mathbf{x}^t$ and the evaluated auxiliary variable $\hat{\mathbf{x}}^{t+1}$, i.e.,

$$\mathbf{x}^{t+1} = \eta^t \hat{\mathbf{x}}^{t+1} + (1 - \eta^t)\mathbf{x}^t. \tag{11}$$

From here, it remains to show how the individual variable copies $\mathbf{z}_i^t$ and Hessian approximations $\mathbf{B}_i^t$ are updated. We use a cyclic update scheme to remove the need to compute all gradients and Hessian approximations at each iteration. Therefore, if we define i_t as the index of the function chosen at step t, it is updated by cyclically iterating through all indices in order, i.e., $i_t = \{1, 2, \ldots, n, 1, 2, \ldots\}$. At time t, we update the variable copies as

$$\mathbf{z}_{i_t}^{t+1} = \mathbf{x}^{t+1}, \quad \mathbf{z}_i^{t+1} = \mathbf{z}_i^t \ \text{ for all } i \neq i_t. \tag{12}$$

Thus only a single variable $\mathbf{z}_{i_t}^{t+1}$ is changed at time t while all others are kept the same. Note that the variable differences in (6) will be null unless $i = i_t$ and thus only one approximation $\mathbf{B}_{i_t}^t$ will change at each time t.

To see that this updating scheme requires evaluation of only a single gradient and Hessian approximation matrix per iteration, consider writing the update in (17) as

$$\hat{\mathbf{x}}^{t+1} = (\tilde{\mathbf{B}}^t)^{-1}\left(\mathbf{u}^t - \mathbf{g}^t\right), \tag{13}$$

where we define as the aggregate Hessian approximation $\tilde{\mathbf{B}}^t := \sum_{i=1}^n \mathbf{B}_i^t$, the aggregate Hessian-variable product $\mathbf{u}^t := \sum_{i=1}^n \mathbf{B}_i^t \mathbf{z}_i^t$, and the aggregate gradient $\mathbf{g}^t := \sum_{i=1}^n \nabla f_i(\mathbf{z}_i^t)$. Then, given that at step t only a single index is updated, we can evaluate these variables for step $t+1$ as

$$\tilde{\mathbf{B}}^{t+1} = \tilde{\mathbf{B}}^t + \left(\mathbf{B}_{i_t}^{t+1} - \mathbf{B}_{i_t}^t\right), \tag{14}$$

$$\mathbf{u}^{t+1} = \mathbf{u}^t + \left(\mathbf{B}_{i_t}^{t+1}\mathbf{z}_{i_t}^{t+1} - \mathbf{B}_{i_t}^t \mathbf{z}_{i_t}^t\right), \tag{15}$$

$$\mathbf{g}^{t+1} = \mathbf{g}^t + \left(\nabla f_{i_t}(\mathbf{z}_{i_t}^{t+1}) - \nabla f_{i_t}(\mathbf{z}_{i_t}^t)\right). \tag{16}$$

Thus, only a single $\mathbf{B}_{i_t}^{t+1}$ and $\nabla f_{i_t}(\mathbf{z}_{i_t}^{t+1})$ need to be computed at each iteration, significantly reducing the computation burden of the algorithm relative to traditional quasi-Newton methods.

Although the update of the matrix $\tilde{\mathbf{B}}^t$ in (7) reduces computational complexity of the IQN method, the update in (13) requires computation of the inverse matrix $(\tilde{\mathbf{B}}^t)^{-1}$ which has a expensive computational complexity of the order $\mathcal{O}(p^3)$. This extra computation cost can be avoided using the fact that the update in (7) can be simplified as

$$\tilde{\mathbf{B}}^{t+1} = \tilde{\mathbf{B}}^t + \frac{\mathbf{y}_{i_t}^t \mathbf{y}_{i_t}^{tT}}{\mathbf{y}_i^{tT}\mathbf{s}_t^{i_t}} - \frac{\mathbf{B}_{i_t}^t \mathbf{s}_{i_t}^t \mathbf{s}_{i_t}^{tT} \mathbf{B}_{i_t}^t}{\mathbf{s}_{i_t}^{tT}\mathbf{B}_{i_t}^t\mathbf{s}_{i_t}^t}. \tag{17}$$

To derive the expression in (17) we have substituted the difference $\mathbf{B}_{i_t}^{t+1} - \mathbf{B}_{i_t}^t$ by its rank two expression in (7). By applying the Sherman-Morrison formula twice to the update in (17) we can directly compute the inverse matrix $(\tilde{\mathbf{B}}^{t+1})^{-1}$ as

$$(\tilde{\mathbf{B}}^{t+1})^{-1} = \mathbf{U}^t + \frac{\mathbf{U}^t(\mathbf{B}_{i_t}^t \mathbf{s}_{i_t}^t)(\mathbf{B}_{i_t}^t \mathbf{s}_{i_t}^t)^T \mathbf{U}^t}{\mathbf{s}_{i_t}^{t\,T}\mathbf{B}_{i_t}^t\mathbf{s}_{i_t}^t - (\mathbf{B}_{i_t}^t \mathbf{s}_{i_t}^t)^T \mathbf{U}^t (\mathbf{B}_{i_t}^t \mathbf{s}_{i_t}^t)}, \tag{18}$$

Algorithm 1 Incremental Quasi-Newton (IQN) method

Require: $\mathbf{x}^0$,$\{\nabla f_i(\mathbf{x}^0)\}_{i=1}^n$, $\{\mathbf{B}_i^0\}_{i=1}^n$, η^t

1: Set $\mathbf{z}_1^0 = \cdots = \mathbf{z}_n^0 = \mathbf{x}^0$

2: Set $(\tilde{\mathbf{B}}^0)^{-1} = (\sum_{i=1}^n \mathbf{B}_i^0)^{-1}$, $\mathbf{u}^0 = \sum_{i=1}^n \mathbf{B}_i^0 \mathbf{x}^0$, $\mathbf{g}^0 = \sum_{i=1}^n \nabla f_i(\mathbf{x}^0)$

3: **for** $t = 0, 1, 2, \ldots$ **do**

4: Set $i_t = (t \mod n) + 1$

5: Compute $\hat{\mathbf{x}}^{t+1} = (\tilde{\mathbf{B}}^t)^{-1}\left(\mathbf{u}^t - \mathbf{g}^t\right)$ [cf. (13)]

6: Update $\mathbf{x}^{t+1} = \eta^t \hat{\mathbf{x}}^{t+1} + (1-\eta^t)\mathbf{x}^t$ [cf. (11)]

7: Compute $\mathbf{s}_{i_t}^{t+1}$, $\mathbf{y}_{i_t}^{t+1}$ [cf. (6)], and $\mathbf{B}_{i_t}^{t+1}$ [cf. (7)]

8: Set $\mathbf{z}_{i_t}^{t+1} = \mathbf{x}^{t+1}$, and $\mathbf{z}_i^{t+1} = \mathbf{z}_i^t$ for $i \neq i_t$ [cf. (12)]

9: Update $\mathbf{u}^{t+1}$ [cf. (15)], $\mathbf{g}^{t+1}$ [cf. (16)], and $(\tilde{\mathbf{B}}^{t+1})^{-1}$ [cf. (18)]

10: **end for**

where the matrix $\mathbf{U}^t$ can be evaluated as

$$\mathbf{U}^t = (\tilde{\mathbf{B}}^t)^{-1} - \frac{(\tilde{\mathbf{B}}^t)^{-1}\mathbf{y}_{i_t}^t \mathbf{y}_{i_t}^{tT}(\tilde{\mathbf{B}}^t)^{-1}}{\mathbf{y}_{i_t}^{tT}\mathbf{s}_{i_t}^t + \mathbf{y}_{i_t}^{tT}(\tilde{\mathbf{B}}^t)^{-1}\mathbf{y}_{i_t}^t}. \tag{19}$$

Note that the computations in (18) and (19) have computational complexity of the order $\mathcal{O}(p^2)$ which is significantly lower than the $\mathcal{O}(p^3)$ cost of computing the inverse directly.

The complete IQN algorithm is outlined in Algorithm 1. Beginning with initial variable $\mathbf{x}^0$ and gradient and Hessian estimates $\nabla f_i(\mathbf{x}^0)$ and $\mathbf{B}_i^0$ for all i, each variable copy $\mathbf{z}_i^0$ is set to $\mathbf{x}^0$ in Step 1 and initial values are set for $\mathbf{u}^0$, $\mathbf{g}^0$ and $(\tilde{\mathbf{B}}^0)^{-1}$ in Step 2. For all t, in Step 4 the index i_t of the next function to update is selected cyclically. Then, $\hat{\mathbf{x}}^{t+1}$ and the next variable $\mathbf{x}^{t+1}$ are computed using (13) and (11) in Steps 5 and 6, respectively. The new BFGS variables $\mathbf{s}_{i_t}^{t+1}$, $\mathbf{y}_{i_t}^{t+1}$, and $\mathbf{B}_{i_t}^{t+1}$ are updated in Step 7. In Step 8, the new variable $\mathbf{z}_{i_t}^{t+1}$ is updated as $\mathbf{x}^{t+1}$, keeping other $\mathbf{z}_i^{t+1}$ the same. Finally, the BFGS variables are used in Step 9 to update $\mathbf{u}^{t+1}$, $\mathbf{g}^{t+1}$, and $(\tilde{\mathbf{B}}^{t+1})^{-1}$. We proceed to analyze the local convergence properties of the IQN method.

4. CONVERGENCE ANALYSIS

We make two assumptions in our analysis of IQN, both of which are standard for second-order optimization methods. The first assumption establishes both strong convexity and Lipschitz continuous gradients for each component function f_i.

Assumption 1 *There exist positive constants $0 < \mu \leq L$ such that, for all $i \in \{1, \ldots, n\}$ and $\mathbf{x}, \tilde{\mathbf{x}} \in \mathbb{R}^p$,*

$$\mu\|\mathbf{x} - \tilde{\mathbf{x}}\| \leq \|\nabla f_i(\mathbf{x}) - \nabla f_i(\tilde{\mathbf{x}})\| \leq L\|\mathbf{x} - \tilde{\mathbf{x}}\|. \tag{20}$$

Thus, each function f_i is strongly convex with parameter μ and has Lipschitz continuous gradients with parameter L. We note that while in some machine learning applications, the loss functions used are not strongly convex, e.g. logistic regression, they can typically be made strongly convex by adding a regularization term to the objective function. Our second assumption is that the component Hessians are also Lipschitz continuous. This assumption is commonly made to prove quadratic convergence of Newton's method [28] and superlinear convergence of quasi-Newton methods [20–22].

Assumption 2 *There exists positive constant $0 < \tilde{L}$ such that, for all i,*

$$\|\nabla^2 f_i(\mathbf{x}) - \nabla^2 f_i(\mathbf{y})\| \leq \tilde{L}\|\mathbf{x} - \mathbf{y}\|. \tag{21}$$

We proceed in our analysis of the convergence properties of IQN, first by establishing a local linear convergence rate, then demonstrating some limit properties of the Hessian approximations and finally by showing that an improved superlinear convergence rate of the sequence of residuals can be obtained. We consider the stepsize $\eta^t = 1$ in our analysis since we focus on local convergence of the proposed IQN method.

To start the analysis, first in the following lemma we show that the residual $\|\mathbf{x}^{t+1} - \mathbf{x}^*\|$ is bounded above by linear and quadratic terms of the average error of the last n iterates.

Lemma 1 *Consider the proposed Incremental Quasi-Newton method (IQN) in* (6)-(18)*. If the conditions in Assumption 1 hold, then the sequence of iterates generated by IQN satisfies the inequality*

$$\|\mathbf{x}^{t+1} - \mathbf{x}^*\| \tag{22}$$
$$\leq \frac{L\Gamma^t}{n}\sum_{i=1}^n \left\|\mathbf{z}_i^t - \mathbf{x}^*\right\|^2 + \frac{\Gamma^t}{n}\sum_{i=1}^n \left\|\left(\mathbf{B}_i^t - \nabla^2 f_i(\mathbf{x}^*)\right)\left(\mathbf{z}_i^t - \mathbf{x}^*\right)\right\|,$$

where $\Gamma^t := \|((1/n)\sum_{i=1}^n \mathbf{B}_i^t)^{-1}\|$.

Lemma 1 shows that the residual $\|\mathbf{x}^{t+1} - \mathbf{x}^*\|$ is upper bounded by a sum of quadratic and linear terms of the last n residuals. This can eventually lead to a superlinear convergence rate by establishing the linear term converges to zero at a fast rate, leaving us with an upper bound of quadratic terms only. First, however, we establish a local linear convergence rate in the proceeding theorem.

Theorem 1 *Consider the proposed Incremental Quasi-Newton method (IQN) in* (6)-(18) *and assume that Assumption 1 holds. Then, for each $r \in (0, 1)$, there are positive constants $\epsilon(r)$ and $\delta(r)$ such that for $\|\mathbf{x}^0 - \mathbf{x}^*\| \leq \epsilon(r)$ and $\|\mathbf{B}_i^0 - \nabla^2 f_i(\mathbf{x}^*)\| \leq \delta(r)$, for all $i = 1, \ldots, n$, the sequence of iterates generated by IQN satisfies*

$$\|\mathbf{x}^t - \mathbf{x}^*\| \leq r^{1+\left[\frac{t-1}{n}\right]^+}\|\mathbf{x}^0 - \mathbf{x}^*\|, \tag{23}$$

where $[\cdot]^+$ refers to the floor function. Moreover, the norms $\|\mathbf{B}_i^t\|$ and $\|(\mathbf{B}_i^t)^{-1}\|$ are uniformly bounded.

From the result in Theorem 1 that the norms $\|\mathbf{B}_i^t\|$ and $\|(\mathbf{B}_i^t)^{-1}\|$ are uniformly bounded, we obtain that Γ^t is uniformly bounded. Moreover, the result in (23) shows that the sequence of iterates $\mathbf{x}^t$ converges linearly to the optimal argument. As a result of this linear convergence we obtain that the sequence of $\|\mathbf{x}^{tn+i} - \mathbf{x}^*\|$ is summable for all i, i.e., $\sum_{t=0}^{\infty}\|\mathbf{x}^{tn+i} - \mathbf{x}^*\| < \infty$. The summability condition is necessary to show that the iterates of BFGS method satisfy the the Dennis-Moré condition in (5). Likewise, in the following proposition, we use the condition $\sum_{t=0}^{\infty}\|\mathbf{x}^{tn+i} - \mathbf{x}^*\| < \infty$ to show that a similar result holds for the iterates of the IQN method.

Proposition 1 *Consider the proposed Incremental Quasi-Newton method (IQN) in* (6)-(18)*. Suppose that the conditions in Theorem 1 are valid and Assumptions 1 and 2 hold. Then, for all $i = 1, \ldots, n$ we have*

$$\lim_{t\to\infty} \frac{\|(\mathbf{B}_i^t - \nabla^2 f_i(\mathbf{x}^*))(\mathbf{z}_i^{t+n} - \mathbf{z}_i^t)\|}{\|\mathbf{z}_i^{t+n} - \mathbf{z}_i^t\|} = 0. \tag{24}$$

Proposition 1 is a necessary step towards a superlinear convergence result because, similarly to the traditional BFGS analysis, it shows that our Hessian approximations do converge to the Hessian over time along the direction pointing towards the optimal point, thus giving us a method that is locally close to Newton's method.

In the following lemma, we use the result in (1) to show that the non-quadratic terms $\|(\mathbf{B}_i^t - \nabla^2 f_i(\mathbf{x}^*))(\mathbf{z}_i^t - \mathbf{x}^*)\|$ in (22) converge to zero faster than $\|\mathbf{z}_i^t - \mathbf{x}^*\|$.

Lemma 2 *Consider the proposed Incremental Quasi-Newton method (IQN) in* (6)-(18)*. Suppose that the conditions in Theorem 1 are valid and Assumptions 1 and 2 hold. Then, the following holds for all $i = 1, \ldots, n$*

$$\lim_{t\to\infty} \frac{\|(\mathbf{B}_i^t - \nabla^2 f_i(\mathbf{x}^*))(\mathbf{z}_i^t - \mathbf{x}^*)\|}{\|\mathbf{z}_i^t - \mathbf{x}^*\|} = 0. \tag{25}$$

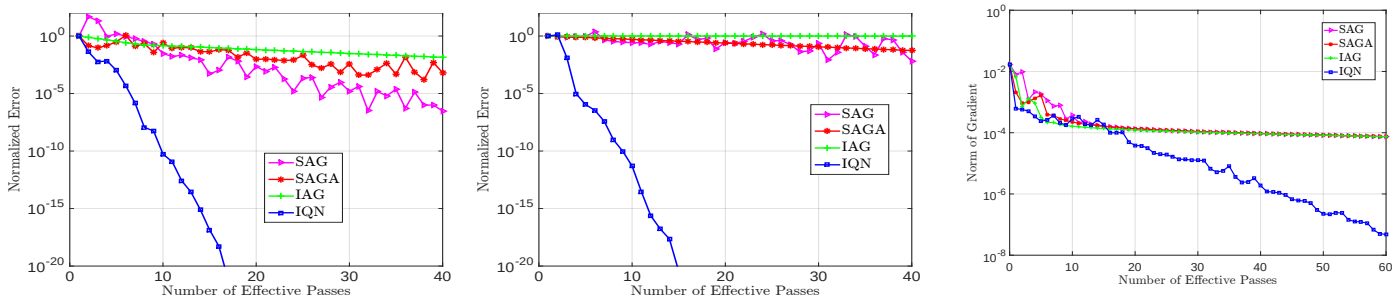

Fig. 1: Convergence results of proposed IQN method in comparison to SAG, SAGA, and IAG. In (a) the left image, we present a sample convergence path of the normalized error on the quadratic program with a small condition number. In (b) the center image, we show the convergence path for the quadratic program with a large condition number. In (c) the right image, we present a sample convergence path for the logistic regression problem. In all cases, IQN provides significant improvement over first order methods, with the difference increasing for larger condition number.

The result in Lemma 2 can thus be used in conjunction with Lemma 1 to show that the residual $\|\mathbf{x}^{t+1}-\mathbf{x}^*\|$ is bounded by a sum of quadratic terms of previous residuals and a term that converges to zero at a fast rate. This result leads us to the main result, namely the local superlinear convergence of the sequence of residuals, stated in the following theorem.

Theorem 2 *Consider the proposed Incremental Quasi-Newton method (IQN) in* (6)-(18)*. Suppose that the conditions in Theorem 1 are valid and Assumptions 1 and 2 hold. Then, the sequence of residuals* $\|\mathbf{x}^t-\mathbf{x}^*\|$ *converges to 0 at a superlinear rate,*

$$\lim_{t\to\infty}\frac{\|\mathbf{x}^{t+1}-\mathbf{x}^*\|}{\|\mathbf{x}^t-\mathbf{x}^*\|}=0. \tag{26}$$

The result in Theorem 2 shows that the sequence of iterates $\mathbf{x}^t$ generated by the IQN method converges superlinealy in a local neighborhood of the optimal argument $\mathbf{x}^*$.

5. NUMERICAL RESULTS

We first provide numerical experiments for the application of IQN in solving the least square problem, or equivalently a quadratic program, in comparison against stochastic and incremental first order methods, namely SAG [12], SAGA [14], and IAG [23]. Consider the problem

$$\mathbf{x}^* := \underset{\mathbf{x}\in\mathbb{R}^p}{\operatorname{argmin}}\ \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\mathbf{x}^T\mathbf{A}_i\mathbf{x}+\mathbf{b}_i^T\mathbf{x}, \tag{27}$$

where $\mathbf{A}_i\in\mathbb{R}^{p\times p}$ is a positive definite matrix and $\mathbf{b}_i\in\mathbb{R}^p$ is a random vector for all i. We select the matrices $\mathbf{A}_i := \text{diag}\{\mathbf{a}_i\}$ randomly for both small (i.e. 10^2) and large (i.e. 10^4) condition numbers while $\mathbf{b}_i$ is chosen uniformly and randomly from the box $[0,10^3]^p$. We set the variable dimension $p=10$ and number of functions $n=1000$. As we are focusing on local convergence, we use a constant step size of $\eta=1$ for the proposed IQN method while choosing the largest step size allowable by the other methods to converge.

In the left image of Figure 1 we present a representative simulation of the convergence path of the normalized error $\|\mathbf{x}^t-\mathbf{x}^*\|/\|\mathbf{x}^0-\mathbf{x}^*\|$ for the quadratic program with small condition number. Step sizes of $\eta=5\times10^{-5}$, $\eta=10^{-4}$ and $\eta=10^{-6}$ were used for SAG, SAGA, and IAG, respectively. These stepsizes are tuned to compare the best performance of these methods with IQN. The proposed method reaches a error of 10^{-10} after 10 passes through the data while SAGA achieves the same error of 10^{-5} after 30 passes (SAG and IAG do not reach 10^{-5} after 40 passes). In the center image, we repeat the same simulation but with larger condition number (SAG using $\eta=2\times10^{-4}$ while others remain the same). Observe that while the performance of IQN does not degrade with larger condition number, the first order methods all suffer large degradation. SAG, SAGA, and IAG reach after 40 passes a normalized error of 6.5×10^{-3}, 5.5×10^{-2}, and 9.6×10^{-1}, respectively. It can be seen that IQN significantly outperforms the first order method for both condition number sizes, with the outperformance increasing for larger condition number. This is expected as first order methods often do not perform well for ill conditioned problems.

5.1. Logistic regression

We evaluate the performance of IQN on a logistic regression problem of practical interest. A logistic regression learns a linear classifier $\mathbf{x}$ that can predict the label of a data point $v_i\in\{-1,1\}$ given a feature vector $\mathbf{u}_i\in\mathbb{R}^p$. In particular, we use IQN and the first order methods to classify two digits from the MNIST handwritten digit database [29]. We evaluate for a set of training samples the probability of a label $v=1$ given an image vector $\mathbf{u}$ as $P(v=1|\mathbf{u})=1/(1+\exp(-\mathbf{u}^T\mathbf{x}))$. The classifier $\mathbf{x}$ is chosen to the vector that maximizes that maximizes the log likelihood over all n samples. Given n images $\mathbf{u}_i$ with associated labels v_i, the optimization problem can be written as

$$\mathbf{x}^* := \underset{\mathbf{x}\in\mathbb{R}^p}{\operatorname{argmin}}\ \frac{\lambda}{2}\|\mathbf{x}\|^2+\frac{1}{n}\sum_{i=1}^{n}\log[1+\exp(-v_i\mathbf{u}_i^T\mathbf{x})], \tag{28}$$

where the first term is a regularization term parametrized by $\lambda\geq0$.

For our simulations we select from the MNIST dataset $n=1000$ images with dimension $p=784$ labelled as one of the digits "0" or "8". The regularization parameter is fixed to be $\lambda=1/n$ and step size of $\eta=0.01$ was used for SAG, SAGA, and IAG. The convergence paths of the norm of the gradient for all methods us shown in the right image in Figure 1. Observe that IQN again outperforms the other stochastic and incremental methods. After 60 passes through the data, IQN reaches a gradient magnitude of 4.8×10^{-8}, while the strongest performing first order method (SAGA) reaches only a magnitude of 7.4×10^{-5}. Additionally, observe that while the first order methods begin to level out after 60 passes, the IQN method continues to descend. This demonstrates the effectiveness of IQN on a practical machine learning problem with real world data.

6. CONCLUSIONS

We presented an incremental quasi-Newton BFGS method to solve a large scale optimization problem, in which an aggregate cost function is minimized while computing only a single gradient and Hessian approximation per iteration. The incremental quasi-Newton (IQN) method iteratively updates approximations to the Hessian inverse to achieve a performance comparable to that of incremental Newton methods. Analytical results were established to show a local superlinear convergence rate and superior numerical performance over first order stochastic and iterative methods for two common machine learning problems.

7. REFERENCES

[1] L. Bottou and Y. L. Cun, "On-line learning for very large datasets," in *Applied Stochastic Models in Business and Industry*, vol. 21. pp. 137-151, 2005.

[2] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *In Proceedings of COMPSTAT'2010*, pp. 177–186, Physica-Verlag HD, 2010.

[3] S. Shalev-Shwartz and N. Srebro, "SVM optimization: inverse dependence on training set size," in *In Proceedings of the 25th international conference on Machine learning*. pp. 928-935, ACM 2008.

[4] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 32–43, 2014.

[5] F. Bullo, J. Cortés, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[6] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 427–438, 2013.

[7] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *Signal Processing, IEEE Transactions on*, vol. 56, no. 7, pp. 3122–3136, 2008.

[8] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links-part i: Distributed estimation of deterministic signals," *Signal Processing, IEEE Transactions on*, vol. 56, no. 1, pp. 350–364, 2008.

[9] A. Ribeiro, "Ergodic stochastic optimization algorithms for wireless communication and networking," *Signal Processing, IEEE Transactions on*, vol. 58, no. 12, pp. 6369–6386, 2010.

[10] ——, "Optimal resource allocation in wireless communication and networking," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–19, 2012.

[11] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in Neural Information Processing Systems*, 2013, pp. 315–323.

[12] N. L. Roux, M. Schmidt, and F. R. Bach, "A stochastic gradient method with an exponential convergence rate for finite training sets," in *Advances in Neural Information Processing Systems*, 2012, pp. 2663–2671.

[13] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Mathematical Programming*, pp. 1–30, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10107-016-1030-6

[14] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Advances in Neural Information Processing Systems*, 2014, pp. 1646–1654.

[15] N. N. Schraudolph, J. Yu, S. Günter *et al.*, "A stochastic quasi-Newton method for online convex optimization." in *AISTATS*, vol. 7, 2007, pp. 436–443.

[16] A. Mokhtari and A. Ribeiro, "RES: Regularized stochastic BFGS algorithm," *Signal Processing, IEEE Transactions on*, vol. 62, no. 23, pp. 6089–6104, 2014.

[17] ——, "Global convergence of online limited memory BFGS," *Journal of Machine Learning Research*, vol. 16, pp. 3151–3181, 2015.

[18] P. Moritz, R. Nishihara, and M. I. Jordan, "A linearly-convergent stochastic L-BFGS algorithm," *arXiv preprint arXiv:1508.02087*, 2015.

[19] R. M. Gower, D. Goldfarb, and P. Richtárik, "Stochastic block BFGS: Squeezing more curvature out of data," *arXiv preprint arXiv:1603.09649*, 2016.

[20] C. G. Broyden, J. E. D. Jr., Wang, and J. J. More, "On the local and superlinear convergence of quasi-Newton methods," *IMA J. Appl. Math*, vol. 12, no. 3, pp. 223–245, June 1973.

[21] M. J. D. Powell, *Some global convergence properties of a variable metric algorithm for minimization without exact line search*, 2nd ed. London, UK: Academic Press, 1971.

[22] J. J. E. Dennis and J. J. More, "A characterization of super linear convergence and its application to quasi-Newton methods," *Mathematics of computation*, vol. 28, no. 126, pp. 549–560, 1974.

[23] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *SIAM Journal on Optimization*, vol. 18, no. 1, pp. 29–51, 2007.

[24] M. Gurbuzbalaban, A. Ozdaglar, and P. Parrilo, "On the convergence rate of incremental aggregated gradient algorithms," *arXiv preprint arXiv:1506.02081*, 2015.

[25] M. Gürbüzbalaban, A. Ozdaglar, and P. Parrilo, "A globally convergent incremental Newton method," *Mathematical Programming*, vol. 151, no. 1, pp. 283–313, 2015.

[26] A. Rodomanov and D. Kropotov, "A superlinearly-convergent proximal newton-type method for the optimization of finite sums," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 2597–2605.

[27] A. Mokhtari, M. Eisen, and A. Ribeiro, "An incremental quasi-Newton method with a local superlinear convergence rate," 2016, available at http://www.seas.upenn.edu/∼maeisen/wiki/incr-bfgs-draft.pdf.

[28] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[29] Y. LeCun, C. Cortes, and C. J. Burges, "MNIST handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.