ON STOCHASTIC AND DETERMINISTIC QUASI-NEWTON METHODS FOR NON-STRONGLY CONVEX OPTIMIZATION: ASYMPTOTIC CONVERGENCE AND RATE ANALYSIS*

FARZAD YOUSEFIAN†, ANGELIA NEDIĆ‡, AND UDAY V. SHANBHAG§

Abstract. Motivated by applications arising from large scale optimization and machine learning, we consider stochastic quasi-Newton (SQN) methods for solving unconstrained convex optimization problems. Much of the convergence analysis of SQN methods, in both full and limited-memory regimes, requires the objective function to be strongly convex. However, this assumption is fairly restrictive and does not hold in many applications. To the best of our knowledge, no rate statements currently exist for SQN methods in the absence of such an assumption. Also, among the existing first-order methods for addressing stochastic optimization problems with merely convex objectives, those equipped with provable convergence rates employ averaging. However, this averaging technique has a detrimental impact on inducing sparsity. Motivated by these gaps, we consider optimization problems with non-strongly convex objectives with Lipschitz but possibly unbounded gradients. The main contributions of the paper are as follows: (i) To address large scale stochastic optimization problems, we develop an iteratively regularized stochastic limited-memory BFGS (IRS-LBFGS) algorithm, where the stepsize, regularization parameter, and the Hessian inverse approximation are updated iteratively. We establish convergence of the iterates (with no averaging) to an optimal solution of the original problem both in an almost-sure sense and in a mean sense. The convergence rate is derived in terms of the objective function values and is shown to be $\mathcal{O}\left(1/k^{(1/3-\epsilon)}\right)$, where ϵ is an arbitrary small positive scalar; (ii) In deterministic regimes, we show that the algorithm displays a rate $\mathcal{O}(1/k^{1-\epsilon})$. We present numerical experiments performed on a large-scale text classification problem and compare IRS-LBFGS with standard SQN methods as well as first-order methods such as SAGA and IAG.

Key words. stochastic optimization, quasi-Newton, regularization, large scale optimization

AMS subject classifications. 65K05, 90C06, 90C30, 90C53

1. Introduction. We consider the following stochastic optimization problem:

$$\text{(SO)} \qquad \min_{x \in \mathbb{R}^n} f(x) \triangleq \mathsf{E}\left[F(x, \xi(\omega))\right],$$

where $F : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$ is a real-valued function, the random vector ξ is defined as $\xi : \Omega \to \mathbb{R}^d$, $(\Omega, \mathcal{F}, \mathbb{P})$ denotes the associated probability space, and the expectation $\mathsf{E}[F(x, \xi)]$ is taken with respect to $\mathbb{P}$. Problem (SO) provides a general framework that can capture a wide range of applications in operations research, machine learning, statistics and control to name a few (cf. [2, 3]). Addressing problem (SO) has led to significant progress via Monte-Carlo sampling techniques. Amongst such schemes, stochastic approximation (SA) methods [22, 25] have proved particularly popular. The standard SA method, introduced by Robbins and Monro [25], for solving (SO), produces a sequence $\{x_k\}$ using the following update rule

$$\text{(SA)} \qquad x_{k+1} := x_k - \gamma_k \nabla F(x_k, \xi_k), \quad \text{for } k \geq 0,$$

where $x_0 \in \mathbb{R}^n$ is a randomly generated initial point, $\gamma_k > 0$ denotes the stepsize and $\nabla F(x_k, \xi_k)$ denotes a sampled gradient of f with respect to x at x_k. SA schemes are characterized by several disadvantages, including a poorer rate of convergence (than their deterministic counterparts) and the detrimental impact of conditioning on their performance. In deterministic regimes, the BFGS method, named after Broyden, Fletcher, Goldfarb, and Shanno, is amongst the most popular quasi-Newton methods [6, 8], displaying a superlinear convergence rate without requiring second-order information. Addressing large scale deterministic problems, the limited-memory variant of the BFGS method, denoted by LBFGS, was developed and attains an R-linear convergence rate under strong convexity of the objective function (see Theorem 6.1 in [13]). Recently, there has been a growing interest in applying stochastic quasi-Newton (SQN) methods for solving large-scale optimization and machine learning problems. In these methods, x_k is updated by the following rule:

(SQN) $$x_{k+1} := x_k - \gamma_k H_k \nabla F(x_k, \xi_k), \quad \text{for } k \geq 0,$$

where $H_k \succeq 0$ is an approximation of the inverse of the Hessian at iteration k that incorporates the curvature information of the objective function within the algorithm. The convergence of this class of algorithms can be derived under a careful choice of H_k and the stepsize sequence $\{\gamma_k\}$. In particular, the boundedness of the eigenvalues of H_k is an important factor in achieving global convergence in convex and nonconvex problems (cf. [1, 12]). While in [27] the performance of SQN methods was found to be favorable in solving high dimensional problems, Mokhtari et al. [19] considered stochastic optimization problems with strongly convex objectives and developed a regularized BFGS method (RES) by updating H_k according to a modified version of BFGS update rule to assure convergence. To address large scale applications, limited-memory variants were employed to ascertain scalability in terms of the number of variables [3, 20]. Recent extensions have included a stochastic quasi-Newton method [30] for solving nonconvex stochastic optimization problems and a constant stepsize variance reduced SQN method [14] for smooth strongly convex problems characterized by a linear convergence rate. Finally, an incremental quasi-Newton (IQN) method with a local superlinear convergence rate has been recently developed for addressing the sum of a large number of strongly convex functions [18].

Motivation: In both the full and limited memory variants of the SQN methods in the literature [3, 18, 20], it is uniformly assumed that the objective function is strongly convex. This assumption plays an important role in deriving the rate of convergence of the algorithm. However, in many applications, the objective function is convex, but not strongly convex such as, when considering the logistic regression function. While a lack of strong convexity might lead to slower convergence in practice, no rigorous support for the convergence rate is currently available in the literature of SQN methods. A simple remedy to address this challenge is to regularize the objective function with the term $\frac{1}{2}\mu\|x\|^2$ and solve the approximate problem of the form $\min_{x \in \mathbb{R}^n} f(x) + \frac{\mu}{2}\|x\|^2$, where $\mu > 0$ is the regularization parameter. Several challenges arise in applying this technique. A drawback of this technique is that the optimal solution to the regularized problem is not necessarily an optimal of the original problem (SO). Yet, another challenge arises from the choice of μ. While larger values of μ may result in large deviations from the true optimal solution(s), choosing a small μ leads to a deterioration of the constant factor in the convergence rate of the algorithm. This issue has been addressed to some extent with the help of averaging techniques. In particular, under mere convexity, most first-order methods admit convergence rate guarantees under averaging. For example, averaging SA schemes achieve a rate of

$\mathcal{O}\left(\frac{M}{\sqrt{k}}\right)$, where M is an upper bound on the norm of the subgradient (see [21,22]). In past few years, fast incremental gradient methods with improved rates of convergence have been developed (see [5,9,26,29]). Of these, addressing the merely convex case, SAGA with averaging achieves a sublinear convergence rate $\mathcal{O}\left(\frac{N}{k}\right)$ where N denotes the number of blocks, while in the presence of strong convexity, non-averaging variants of SAGA and IAG admit a linear convergence rate assuming that the function satisfies some smoothness conditions.

A crucial concern that plagues the aforementioned schemes is that the averaging technique has a detrimental impact on inducing sparsity. In the case of incremental methods such as SAGA and IAG, despite the fast convergence speed, the application of these methods is impaired by the excessive memory requirements. For standard SAGA and IAG, the memory requirements are $\mathcal{O}(nN)$. Accordingly, in this paper, our main goal lies in addressing such shortcomings in absence of strong convexity and developing a first-order method equipped with a rate of convergence for the generated non-averaged iterates.

Related research on regularization: In optimization, in order to obtain solutions with desirable properties, it is common to regularize the problem (SO) as follows

$$(1) \qquad \min_{x \in \mathbb{R}^n} f_\mu(x) := f(x) + \mu R(x),$$

where $R : \mathbb{R}^n \to \mathbb{R}$ is a proper convex function and the scalar $\mu > 0$ is the regularization parameter. The properties of the regularized problem and its relation to the original problem have been investigated by different researchers. Mangasarian and his colleagues appear among the first researchers who studied exact regularization of linear and nonlinear programs [15,17]. A regularization is said to be exact when an optimal solution of (1), is also optimal for problem (SO) if μ is small enough. Tseng et al. [7,28] established the necessary and sufficient conditions of exact regularization for convex programs and derived error bounds for inexact regularized convex problems. In a similar veing, exact regularization of variational inequality problems has been studied in [4]. A challenging question is concerned about the choice of the regularization parameter μ. A common approach to find an acceptable value for μ is through a two-loop scheme where in the inner loop, problem (1) is solved for a fixed value of μ, while μ is tuned in the outer loop. The main drawback of this approach is that, in general, there is no guidance on the tuning rule for μ. In addition, this approach is computationally inefficient. Furthermore, tuning rules may result in losing the desired properties of the sample path of the solutions to regularized problems. In this work, we address this issue through employing an iterative single-loop algorithm where we update the regularization parameter μ at each iteration of the scheme and reduce it iteratively to converge to zero [10,34].

Contributions: We consider stochastic optimization problems with non-strongly convex objective functions and Lipschitz but possibly unbounded gradient mappings. Our main contributions are as follows:

(i) **Asymptotic convergence**: We develop an iteratively regularized SQN method where the stepsize, regularization parameter, and the Hessian inverse approximation denoted by H_k are updated iteratively. We assume that H_k satisfies a set of general assumptions on its eigenvalues and its dependency on the uncertainty. The asymptotic convergence of the method is established under a suitable choice of an error function. For the sequence of the iterates $\{x_k\}$ produced by the algorithm, we obtain a set of suitable conditions on the stepsize and regularization sequences for which $f(x_k)$ converges to the optimal objective value, i.e., f^*, of (SO) both in an almost

sure sense and in a mean sense. We also derive an upper bound for $f(x_k) - f^*$.

(ii) **Rate of convergence for regularized LBFGS methods**: To address large scale stochastic optimization problems, motivated by our earlier work [33] on SQN methods for small scale stochastic optimization problems with non-strongly convex objectives, we develop an iteratively regularized stochastic limited-memory BFGS scheme (see Algorithm 1). We show that under a careful choice of the update rules for the stepsize and regularization parameter, Algorithm 1 displays a convergence rate $\mathcal{O}\left(k^{-\left(\frac{1}{3} - \epsilon\right)}\right)$ in terms of the objective function values, where ϵ is an arbitrary small positive scalar. Similar to standard stochastic LBFGS schemes, the memory requirement is independent of N and is $\mathcal{O}(mn)$, where $m \ll n$ denotes the memory parameter in the LBFGS scheme. In the deterministic case, we show that the convergence rate improves to $\mathcal{O}\left(\frac{1}{k^{1-\epsilon}}\right)$. Both of these convergence rates appear to be new for the class of deterministic and stochastic quasi-Newton methods.

Outline of the paper: The rest of the paper is organized as follows. Section 2 presents the general framework of the proposed SQN algorithm and the sets of main assumptions. In Section 3, we prove the asymptotic convergence of the iterates produced by the scheme in both almost sure and a mean sense and derive the a general error bound. In Section 4, we develop an iteratively regularized stochastic LBFGS method (Algorithm 1) and derive its convergence rate. The rate analysis is also provided for the deterministic variant of this scheme. We then present the numerical experiments performed on a large scale classification problem in Section 5. The paper ends with some concluding remarks in Section 6.

Notation: A vector x is assumed to be a column vector and x^T denotes its transpose, while $\|x\|$ denotes the Euclidean vector norm, i.e., $\|x\| = \sqrt{x^T x}$. We write *a.s.* as the abbreviation for "almost surely". For a symmetric matrix B, $\lambda_{\min}(B)$ and $\lambda_{\max}(B)$ denotes the smallest and largest eigenvalue of B, respectively. We use $\mathsf{E}[z]$ to denote the expectation of a random variable z. A function $f : X \subset \mathbb{R}^n \to \mathbb{R}$ is said to be strongly convex with parameter $\mu > 0$, if $f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\mu}{2}\|x-y\|^2$, for any $x, y \in X$. A mapping $F : X \subset \mathbb{R}^n \to \mathbb{R}$ is said to be Lipschitz continuous with parameter $L > 0$ if for any $x, y \in X$, we have $\|F(x) - F(y)\| \leq L\|x - y\|$. For a continuously differentiable function f with Lipschitz gradients with parameter $L > 0$, we have $f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{L}{2}\|x - y\|^2$, for any $x, y \in X$. For a vector $x \in \mathbb{R}^n$ and a nonempty set $X \subset \mathbb{R}^n$, the Euclidean distance of x from X is denoted by $dist(x, X)$. We denote the optimal objective value of problem (SO) by f^* and the set of the optimal solutions by X^*.

2. Outline of the SQN scheme. In this section, we describe a general SQN scheme for solving problem (SO) and present the main assumptions. Let $x_0 \in \mathbb{R}^n$ be an arbitrary initial point, and x_k be generated by the following recursive rule

(IR-SQN) $\qquad x_{k+1} := x_k - \gamma_k H_k \left(\nabla F(x_k, \xi_k) + \mu_k(x_k - x_0)\right), \quad$ for all $k \geq 0$.

Here, γ_k and μ_k are the steplength and the regularization parameter, respectively. $H_k \in \mathbb{R}^{n \times n}$ is a matrix that contains the curvature information of the objective function. The (IR-SQN) scheme can be seen as a regularized variant of the classical stochastic SQN method. Here we regularize the gradient map by the term $\mu_k(x_k - x_0)$ to induce the strong monotonicity property. In the absence of strong convexity of f, unlike the classical schemes where μ_k is maintained fixed, we let μ_k be updated and decreased to zero. Throughout, we let $\mathcal{F}_k$ denote the history of the method up to time k, i.e., $\mathcal{F}_k \triangleq \{x_0, \xi_0, \xi_1, \ldots, \xi_{k-1}\}$ for $k \geq 1$, and $\mathcal{F}_0 \triangleq \{x_0\}$.

ASSUMPTION 1. *Consider problem* (SO). *Let the following hold:*
 (a) The function $f(x)$ is convex over $\mathbb{R}^n$.
 (b) $f(x)$ is continuously differentiable with Lipschitz continuous gradients over $\mathbb{R}^n$ with parameter $L > 0$.
 (c) The optimal solution set of the problem is nonempty.

Next, we state the assumptions on the random variable ξ and the properties of the stochastic estimator of the gradient mapping.

ASSUMPTION 2 (Random variable ξ).
 (a) Vectors $\xi_k \in \mathbb{R}^d$ are i.i.d. realizations of the random variable ξ for any $k \geq 0$;
 (b) The stochastic gradient mapping $\nabla F(x, \xi)$ is an unbiased estimator of $\nabla f(x)$, i.e. $\mathsf{E}[\nabla F(x, \xi)] = \nabla f(x)$ for all x, and has a bounded variance, i.e., there exists a scalar $\nu > 0$ such that $\mathsf{E}[\|\nabla F(x, \xi) - \nabla f(x)\|^2] \leq \nu^2$, for all $x \in \mathbb{R}^n$.

The next assumption pertains to the properties of H_k.

ASSUMPTION 3 (Conditions on matrix H_k). *Let the following hold for all $k \geq 0$:*
(a) The matrix $H_k \in \mathbb{R}^{n \times n}$ is $\mathcal{F}_k$-measurable, i.e., $\mathsf{E}[H_k \mid \mathcal{F}_k] = H_k$.
(b) Matrix H_k is symmetric and positive definite and satisfies the following condition: There exist positive scalars $\lambda_{\min}, \lambda$ and scalar $\alpha \leq 0$ such that

$$\lambda_{\min}\mathbf{I} \preceq H_k \preceq \lambda\mu_k^\alpha\mathbf{I}, \qquad \text{for all } k \geq 0,$$

where μ_k is the regularization parameter in (IR-SQN).

Assumption 3 holds for the stochastic gradient method where H_k is the identity matrix, $\lambda_{\min} = \lambda = 1$ and $\alpha = 0$. In the case of employing an appropriate LBFGS update rule that will be discussed in Section 4, the maximum eigenvalue is obtained in terms of the regularization parameter.

3. Convergence analysis. In this section, we present the convergence analysis of the (IR-SQN) method. Our discussion starts by some preliminary definitions and properties. After obtaining a recursive error bound for the method in Lemma 3, we show a.s. convergence in Proposition 4, establish convergence in mean, and derive an error bound in Proposition 5.

DEFINITION 1 (Regularized function and gradient mapping). *Consider the sequence $\{\mu_k\}$ of positive scalars and the starting point of the algorithm* (IR-SQN), *i.e., x_0. The regularized function f_k and its gradient are defined as follows for all $k \geq 0$:*

$$f_k(x) \triangleq f(x) + \frac{\mu_k}{2}\|x - x_0\|^2, \quad \nabla f_k(x) \triangleq \nabla f(x) + \mu_k(x - x_0).$$

In a similar way, we denote the regularized stochastic function $F(x, \xi)$ and its gradient with F_k and ∇F_k for any ξ, respectively.

PROPERTY 1 (Properties of a regularized function). *We have:*
 (a) f_k is strongly convex with a parameter μ_k.
 (b) f_k has Lipschitzian gradients with parameter $L + \mu_k$.
 (c) f_k has a unique minimizer over $\mathbb{R}^n$, denoted by x_k^*. Moreover, for any $x \in \mathbb{R}^n$,

$$2\mu_k(f_k(x) - f_k(x_k^*)) \leq \|\nabla f_k(x)\|^2 \leq 2(L + \mu_k)(f_k(x) - f_k(x_k^*)).$$

The existence and uniqueness of x_k^* in Property 1(c) is due to the strong convexity of the function f_k (see, for example, Section 1.3.2 in [24]), while the relation for the gradient is known to hold for a strongly convex function with a parameter μ that also

has Lipschitz gradients with a parameter L (see Lemma 1 on page 23 in [24]). In the convergence analysis, we make use of the following result, which can be found in [24] (see Lemma 10 on page 49).

LEMMA 2. *Let $\{v_k\}$ be a sequence of nonnegative random variables, where $\mathsf{E}[v_0] < \infty$, and let $\{\alpha_k\}$ and $\{\beta_k\}$ be deterministic scalar sequences such that:*

$$\mathsf{E}[v_{k+1}|v_0,\ldots,v_k] \le (1-\alpha_k)v_k + \beta_k \quad a.s. \text{ for all } k \ge 0,$$

where $0 \le \alpha_k \le 1$, $\beta_k \ge 0$, $\sum_{k=0}^{\infty} \alpha_k = \infty$, $\sum_{k=0}^{\infty} \beta_k < \infty$, and $\lim_{k\to\infty} \frac{\beta_k}{\alpha_k} = 0$. Then, $v_k \to 0$ almost surely.

Throughout, we denote the stochastic error of the gradient estimator by

$$(2) \qquad\qquad w_k \triangleq \nabla F(x_k, \xi_k) - \nabla f(x_k), \quad \text{for all } k \ge 0.$$

Note that under Assumption 2, from the definition of w_k in (2), we obtain $\mathsf{E}[w_k \mid \mathcal{F}_k] = 0$ and $\mathsf{E}\big[\|w_k\|^2 \mid \mathcal{F}_k\big] \le \nu^2$. The following result plays a key role in the convergence and rate analysis of the proposed schemes.

LEMMA 3 (A recursive error bound). *Consider the (IR-SQN) method and suppose Assumptions 1, 2, and 3 hold. Also, assume μ_k is a non-increasing sequence and let*

$$(3) \qquad\qquad \gamma_k \mu_k^{2\alpha} \le \frac{\lambda_{\min}}{\lambda^2(L+\mu_0)}, \quad \text{for all } k \ge 0.$$

Then, the following inequality holds for all $k \ge 0$:

$$\mathsf{E}[f_{k+1}(x_{k+1}) \mid \mathcal{F}_k] - f^* \le (1 - \lambda_{\min}\mu_k\gamma_k)(f_k(x_k) - f^*) + \frac{\lambda_{\min}\,dist^2(x_0, X^*)}{2}\mu_k^2\gamma_k$$

$$(4) \qquad\qquad\qquad + \frac{(L+\mu_k)\lambda^2\nu^2}{2}\mu_k^{2\alpha}\gamma_k^2.$$

Proof. The Lipschitzian property of ∇f_k and the update rule (IR-SQN) imply that

$$f_k(x_{k+1}) \le f_k(x_k) + \nabla f_k(x_k)^T(x_{k+1} - x_k) + \frac{(L+\mu_k)}{2}\|x_{k+1} - x_k\|^2$$

$$= f_k(x_k) - \gamma_k \nabla f_k(x_k)^T H_k \left(\nabla F(x_k, \xi_k) + \mu_k(x_k - x_0)\right)$$

$$+ \frac{(L+\mu_k)}{2}\gamma_k^2 \|H_k \left(\nabla F(x_k, \xi_k) + \mu_k(x_k - x_0)\right)\|^2.$$

Invoking the definition of the stochastic error w_k in (2) and Definition 1, we obtain

$$f_k(x_{k+1}) \le f_k(x_k) - \gamma_k \nabla f_k(x_k)^T H_k(\nabla f(x_k) + w_k + \mu_k(x_k - x_0))$$

$$(5) \qquad + \frac{(L+\mu_k)}{2}\gamma_k^2\|H_k(\nabla f(x_k) + w_k + \mu_k(x_k - x_0))\|^2$$

$$= f_k(x_k) - \gamma_k \underbrace{\nabla f_k(x_k)^T H_k(\nabla f_k(x_k) + w_k)}_{\text{Term 1}} + \frac{(L+\mu_k)}{2}\gamma_k^2 \underbrace{\|H_k(\nabla f_k(x_k) + w_k)\|^2}_{\text{Term 2}},$$

where in the last equation, we used the definition of f_k. Next, we estimate the conditional expectation of Term 1 and 2. From Assumption 3, we have

$$\text{Term 1} = \nabla f_k(x_k)^T H_k \nabla f_k(x_k) + \nabla f_k(x_k)^T H_k w_k$$

$$\ge \lambda_{\min}\|\nabla f_k(x_k)\|^2 + \nabla f_k(x_k)^T H_k w_k.$$

Taking expectations conditioned on $\mathcal{F}_k$, from the preceding inequality, we obtain

(6) $\mathsf{E}[\text{Term } 1 \mid \mathcal{F}_k] \geq \lambda_{\min}\|\nabla f_k(x_k)\|^2 + \mathsf{E}\left[\nabla f_k(x_k)^T H_k w_k \mid \mathcal{F}_k\right]$
$$= \lambda_{\min}\|\nabla f_k(x_k)\|^2 + \nabla f_k(x_k)^T H_k \mathsf{E}[w_k \mid \mathcal{F}_k] = \lambda_{\min}\|\nabla f_k(x_k)\|^2,$$

where we recall that $\mathsf{E}[w_k \mid \mathcal{F}_k] = 0$ and $\mathsf{E}[H_k \mid \mathcal{F}_k] = H_k$. Similarly, in Term 2, invoking Assumption 3(b), we may write

$$\text{Term } 2 = (\nabla f_k(x_k) + w_k)^T H_k^2 (\nabla f_k(x_k) + w_k) \leq (\lambda \mu_k^\alpha)^2 \|\nabla f_k(x_k) + w_k\|^2$$
$$= \lambda^2 \mu_k^{2\alpha} \left(\|\nabla f_k(x_k)\|^2 + \|w_k\|^2 + 2\nabla f_k(x_k)^T w_k\right).$$

Taking conditional expectations from the preceding inequality, and using Assumption 2, we obtain

$$\mathsf{E}[\text{Term } 2 \mid \mathcal{F}_k] \leq \lambda^2 \mu_k^{2\alpha} \left(\|\nabla f_k(x_k)\|^2 + \mathsf{E}\left[\|w_k\|^2 \mid \mathcal{F}_k\right] + 2\nabla f_k(x_k)^T \mathsf{E}[w_k \mid \mathcal{F}_k]\right)$$
(7) $$\leq \lambda^2 \mu_k^{2\alpha} \left(\|\nabla f_k(x_k)\|^2 + \nu^2\right).$$

Next, taking conditional expectations in (5), and using (6) and (7), we obtain

$$\mathsf{E}[f_k(x_{k+1}) \mid \mathcal{F}_k] \leq f_k(x_k) - \gamma_k \lambda_{\min}\|\nabla f_k(x_k)\|^2$$
$$+ \lambda^2 \mu_k^{2\alpha} \frac{(L + \mu_k)}{2} \gamma_k^2 \left(\|\nabla f_k(x_k)\|^2 + \nu^2\right)$$
$$\leq f_k(x_k) - \frac{\gamma_k \lambda_{\min}}{2}\|\nabla f_k(x_k)\|^2 \left(2 - \frac{\lambda^2 \mu_k^{2\alpha}\gamma_k(L+\mu_k)}{\lambda_{\min}}\right) + \lambda^2 \mu_k^{2\alpha} \frac{(L+\mu_k)}{2}\gamma_k^2 \nu^2.$$

From the assumption that γ_k and μ_k satisfy $\gamma_k \mu_k^{2\alpha} \leq \frac{\lambda_{\min}}{\lambda^2(L+\mu_0)}$ for any $k \geq 0$ and that μ_k is non-increasing, we have $\gamma_k \mu_k^{2\alpha} \leq \frac{\lambda_{\min}}{\lambda^2(L+\mu_k)}$. As a consequence, we get $2 - \frac{\lambda^2 \mu_k^{2\alpha}\gamma_k(L+\mu_k)}{\lambda_{\min}} \geq 1$. Therefore, from the preceding inequality, we obtain

$$\mathsf{E}[f_k(x_{k+1}) \mid \mathcal{F}_k] \leq f_k(x_k) - \frac{\gamma_k \lambda_{\min}}{2}\|\nabla f_k(x_k)\|^2 + \lambda^2 \mu_k^{2\alpha} \frac{(L+\mu_k)}{2}\gamma_k^2 \nu^2.$$

Employing Property 1(c), we have

$$\mathsf{E}[f_k(x_{k+1}) \mid \mathcal{F}_k] \leq f_k(x_k) - \lambda_{\min}\mu_k\gamma_k(f_k(x_k) - f_k(x_k^*)) + \lambda^2 \mu_k^{2\alpha} \frac{(L+\mu_k)}{2}\gamma_k^2 \nu^2.$$

Note that, since μ_k is a non-increasing sequence, Definition 1 implies that

$$\mathsf{E}[f_{k+1}(x_{k+1}) \mid \mathcal{F}_k] \leq \mathsf{E}[f_k(x_{k+1}) \mid \mathcal{F}_k].$$

Therefore, we obtain

(8)
$$\mathsf{E}[f_{k+1}(x_{k+1}) \mid \mathcal{F}_k] \leq f_k(x_k) - \lambda_{\min}\mu_k\gamma_k\underbrace{(f_k(x_k) - f_k(x_k^*))}_{\text{Term } 3} + \lambda^2 \mu_k^{2\alpha} \frac{(L+\mu_k)}{2}\gamma_k^2 \nu^2.$$

Next, we derive a lower bound for Term 3. Since x_k^* is the unique minimizer of f_k, we have $f_k(x_k^*) \leq f_k(x^*)$. Therefore, invoking Definition 1, for an arbitrary optimal solution $x^* \in X^*$, we have

$$f_k(x_k) - f_k(x_k^*) \geq f_k(x_k) - f_k(x^*) = f_k(x_k) - f^* - \frac{\mu_k}{2}\|x^* - x_0\|^2.$$

7

From the preceding relation and (8), we have

$$\mathsf{E}[f_{k+1}(x_{k+1}) \mid \mathcal{F}_k] \le f_k(x_k) - \lambda_{\min}\mu_k\gamma_k(f_k(x_k) - f^*) + \frac{\lambda_{\min}\|x^* - x_0\|^2}{2}\mu_k^2\gamma_k$$
$$+ \frac{(L + \mu_k)\lambda^2\nu^2}{2}\mu_k^{2\alpha}\gamma_k^2.$$

Since x^* is an arbitrary optimal solution, taking minimum from the right-hand side of the preceding inequality over X^*, we can replace $\|x^* - x_0\|$ by $\mathrm{dist}(x_0, X^*)$. Then, subtracting f^* from both sides of the resulting relation yields the desired inequality.□

Next, we show the convergence of the scheme. In order to apply Lemma 2 to inequality (4) and prove the almost sure convergence, we use the following definitions:

$$v_k := f_k(x_k) - f^*, \quad \alpha_k := \lambda_{\min}\gamma_k\mu_k,$$

(9) $$\beta_k := \frac{\lambda_{\min}\mathrm{dist}^2(x_0, X^*)}{2}\mu_k^2\gamma_k + \frac{(L + \mu_k)\lambda^2\nu^2}{2}\mu_k^{2\alpha}\gamma_k^2.$$

To satisfy the conditions of Lemma 2, we identify a set of sufficient conditions on $\{\gamma_k\}$ and $\{\mu_k\}$ in the following assumption. Later in the subsequent sections, for each class of algorithms, we provide a set of sequences that meet these assumptions.

ASSUMPTION 4 (Conditions on sequences for a.s. convergence). *Let the sequences $\{\gamma_k\}$ and $\{\mu_k\}$ be positive and satisfy the following conditions:*

(a) $\lim_{k\to\infty} \gamma_k\mu_k^{2\alpha-1} = 0;$ (b) $\{\mu_k\}$ *is non-increasing and* $\mu_k \to 0;$

(c) $\lambda_{\min}\gamma_k\mu_k \le 1$ *for* $k \ge 0;$ (d) $\sum_{k=0}^{\infty} \gamma_k\mu_k = \infty;$

(e) $\sum_{k=0}^{\infty} \mu_k^2\gamma_k < \infty;$ (f) $\sum_{k=0}^{\infty} \gamma_k^2\mu_k^{2\alpha} < \infty.$

PROPOSITION 4 (Almost sure convergence). *Consider the* (IR-SQN) *scheme. Suppose Assumptions 1, 2, 3, and 4 hold. Then,* $\lim_{k\to\infty} f(x_k) = f^*$ *almost surely.*

Proof. First, note that from Assumption 4(a,b), we have $\lim_{k\to\infty} \gamma_k\mu_k^{2\alpha} = 0$. Thus, there exists $K \ge 1$ such that for any $k \ge K$, we have $\gamma_k\mu_k^{2\alpha} \le \frac{\lambda_{\min}}{\lambda^2(L+\mu_0)}$ implying that condition (3) of Lemma 3 holds for all $k \ge K$. Hence, relation (4) holds for any $k \ge K$. Next, we apply Lemma 2 to prove a.s. convergence of the (IR-SQN) scheme. Consider the definitions in (9) for any $k \ge K$. The non-negativity of α_k and β_k is implied by the definition and that $\lambda_{\min}, \gamma_k$ and μ_k are positive. From (4), we have

$$\mathsf{E}[v_{k+1} \mid \mathcal{F}_k] \le (1 - \alpha_k)v_k + \beta_k \quad \text{for all } k \ge K.$$

Since $f^* \le f(x)$ for any $x \in \mathbb{R}^n$, we can write $v_k = (f(x_k) - f^*) + \frac{\mu_k}{2}\|x_k - x_0\|^2 \ge 0$. From Assumption 4(c), we obtain $\alpha_k \le 1$. Also, from Assumption 4(d), we get $\sum_{k=K}^{\infty} \alpha_k = \infty$. Using Assumption 4(b,e,f) and the definition of β_k in (9), for an arbitrary solution x^*, we may prove the summability of β_k as follows.

$$\sum_{k=K}^{\infty} \beta_k \le \frac{\lambda_{\min}\mathrm{dist}^2(x_0, X^*)}{2} \sum_{k=K}^{\infty} \mu_k^2\gamma_k + \frac{(L + \mu_0)\lambda^2\nu^2}{2} \sum_{k=K}^{\infty} \mu_k^{2\alpha}\gamma_k^2 < \infty.$$

Similarly, we can write

$$\lim_{k\to\infty} \frac{\beta_k}{\alpha_k} \le \frac{\mathrm{dist}^2(x_0, X^*)}{2} \lim_{k\to\infty} \mu_k + \frac{(L + \mu_0)\lambda^2\nu^2}{2} \lim_{k\to\infty} \mu_k^{2\alpha-1}\gamma_k = 0,$$

8

where the last equation is implied by Assumption 4(a,b). Therefore, all conditions of Lemma 2 hold (with an index shift) and we conclude that $v_k := f_k(x_k) - f^*$ converges to 0 a.s. Let us define $v'_k := f(x_k) - f^*$ and $v''_k := \frac{\mu_k}{2}\|x_k - x_0\|^2$, so that $v_k = v'_k + v''_k$. Since v'_k and v''_k are nonnegative, and $v_k \to 0$ a.s., it follows that $v'_k \to 0$ and $v''_k \to 0$ a.s., implying that $\lim_{k\to\infty} f(x_k) = f^*$ a.s. $\qquad\square$

In the following, our goal is to state the assumptions on the sequences $\{\gamma_k\}$ and $\{\mu_k\}$ under which we can show the convergence in mean.

ASSUMPTION 5 (Conditions on sequences for convergence in mean). *Let the sequences $\{\gamma_k\}$ and $\{\mu_k\}$ be positive and satisfy the following conditions:*
  (a) $\lim_{k\to\infty} \gamma_k \mu_k^{2\alpha-1} = 0$;
  (b) $\{\mu_k\}$ *is non-increasing and* $\mu_k \to 0$;
  (c) $\lambda_{\min}\gamma_k\mu_k \leq 1$ *for* $k \geq 0$;
  (d) There exist K_0 *and* $0 < \beta < 1$ *such that*

$$\gamma_{k-1}\mu_{k-1}^{2\alpha-1} \leq \gamma_k\mu_k^{2\alpha-1}(1 + \beta\lambda_{\min}\gamma_k\mu_k), \quad \text{for all } k \geq K_0;$$

  (e) There exists a scalar $\rho > 0$ *such that* $\mu_k^{2-2\alpha} \leq \rho\gamma_k$ *for all* $k \geq 0$.

Next, we use Assumption 5 to establish the convergence in mean.

PROPOSITION 5 (Convergence in mean). *Consider the* (IR-SQN) *scheme. Suppose Assumptions 1, 2, 3, and 5 hold. Then, there exists* $K \geq 1$ *such that:*

$$(10) \qquad \mathsf{E}[f(x_{k+1})] - f^* \leq \theta\gamma_k\mu_k^{2\alpha-1}, \quad \text{for all } k \geq K,$$

where f^* *is the optimal value of problem and*

$$(11) \qquad \theta := \max\left\{ \frac{\mathsf{E}[f_{K+1}(x_{K+1})] - f^*}{\gamma_K\mu_K^{2\alpha-1}}, \frac{\rho\lambda_{\min}\mathrm{dist}^2(x_0, X^*) + (L+\mu_0)\lambda^2\nu^2}{2\lambda_{\min}(1-\beta)} \right\}.$$

Moreover, $\lim_{k\to\infty} \mathsf{E}[f(x_k)] = f^*$.

Proof. Note that Assumption 5(a,b) imply that (4) holds for a large enough k, say after $\hat{K}$. Then, since the conditions of Lemma 3 are met (with an index shift), taking expectations on both sides of (4), we obtain for any $k \geq \hat{K}$:

$$\mathsf{E}[f_{k+1}(x_{k+1}) - f^*] \leq (1 - \lambda_{\min}\mu_k\gamma_k)\mathsf{E}[f_k(x_k) - f^*] + \frac{\lambda_{\min}\mathrm{dist}^2(x_0, X^*)}{2}\mu_k^2\gamma_k$$
$$+ \frac{(L+\mu_0)\lambda^2\nu^2}{2}\mu_k^{2\alpha}\gamma_k^2.$$

Using Assumption 5(e), we have $\mu_k^2\gamma_k \leq \rho\gamma_k^2\mu_k^{2\alpha}$. Thus, we obtain

$$\mathsf{E}[f_{k+1}(x_{k+1}) - f^*] \leq (1 - \lambda_{\min}\mu_k\gamma_k)\mathsf{E}[f_k(x_k) - f^*]$$
$$(12) \qquad\qquad + \left(\frac{\rho\lambda_{\min}\mathrm{dist}^2(x_0, X^*) + (L+\mu_0)\lambda^2\nu^2}{2} \right)\mu_k^{2\alpha}\gamma_k^2.$$

Let us define $K \triangleq \max\{\hat{K}, K_0\}$, where K_0 is from Assumption 5(d). Using the preceding relation and by induction on k, we show the desired result. To show (10), we show the following relation first:

$$(13) \qquad \mathsf{E}[f_{k+1}(x_{k+1})] - f^* \leq \theta\gamma_k\mu_k^{2\alpha-1}, \quad \text{for all } k \geq K,$$

9

Note that (13) implies the relation (10) since we have $\mathsf{E}[f(x_{k+1})] \leq \mathsf{E}[f_{k+1}(x_{k+1})]$. First, we show that (13) holds for $k = K$. Consider the term $\mathsf{E}[f_{K+1}(x_{K+1})] - f^*$. Multiplying and dividing by $\gamma_K \mu_K^{2\alpha-1}$, we obtain

$$\mathsf{E}[f_{K+1}(x_{K+1})] - f^* = \left(\frac{\mathsf{E}[f_{K+1}(x_{K+1})] - f^*}{\gamma_k \mu_K^{2\alpha-1}} \right) \gamma_K \mu_K^{2\alpha-1} \leq \theta \gamma_K \mu_K^{2\alpha-1},$$

where the last inequality is obtained by invoking the definition of θ in (11). This implies that (13) holds for $k = K$. Now assume that (13) holds for some $k \geq K$. We show that it also holds for $k + 1$. From the induction hypothesis and (12) we have

$$\mathsf{E}[f_{k+1}(x_{k+1}) - f^*] \leq (1 - \lambda_{\min}\mu_k\gamma_k)\theta\gamma_{k-1}\mu_{k-1}^{2\alpha-1}$$
$$+ \left(\frac{\rho\lambda_{\min}\mathrm{dist}^2(x_0, X^*) + (L + \mu_0)\lambda^2\nu^2}{2} \right) \mu_k^{2\alpha}\gamma_k^2.$$

Using Assumption 5(d) we obtain

(14)
$$\mathsf{E}[f_{k+1}(x_{k+1}) - f^*] \leq \theta\gamma_k\mu_k^{2\alpha-1}(1 - \lambda_{\min}\mu_k\gamma_k)(1 + \beta\lambda_{\min}\gamma_k\mu_k)$$
$$+ \left(\frac{\rho\lambda_{\min}\mathrm{dist}^2(x_0, X^*) + (L + \mu_0)\lambda^2\nu^2}{2} \right) \mu_k^{2\alpha}\gamma_k^2.$$

Next we find an upper bound for the term $(1 - \lambda_{\min}\mu_k\gamma_k)(1 + \beta\lambda_{\min}\gamma_k\mu_k)$ as follows

$$(1 - \lambda_{\min}\mu_k\gamma_k)(1 + \beta\lambda_{\min}\gamma_k\mu_k) = 1 - \lambda_{\min}\mu_k\gamma_k + \beta\lambda_{\min}\mu_k\gamma_k - \beta\lambda_{\min}^2\mu_k^2\gamma_k^2$$
$$\leq 1 - (1 - \beta)\lambda_{\min}\mu_k\gamma_k.$$

Combining this relation with (14), it follows that

$$\mathsf{E}[f_{k+1}(x_{k+1}) - f^*] \leq \theta\gamma_k\mu_k^{2\alpha-1} - \theta\lambda_{\min}(1 - \beta)\mu_k^{2\alpha}\gamma_k^2$$
$$+ \left(\frac{\rho\lambda_{\min}\mathrm{dist}^2(x_0, X^*) + (L + \mu_0)\lambda^2\nu^2}{2} \right) \mu_k^{2\alpha}\gamma_k^2$$
$$= \theta\gamma_k\mu_k^{2\alpha-1} - \left(\underbrace{\theta\lambda_{min}(1 - \beta) - \frac{\rho\lambda_{\min}\mathrm{dist}^2(x_0, X^*) + (L + \mu_0)\lambda^2\nu^2}{2}}_{\text{Term 1}} \right) \mu_k^{2\alpha}\gamma_k^2.$$

Note that the definition of θ in (11) implies that Term 1 is nonnegative. Therefore,

$$\mathsf{E}[f_{k+1}(x_{k+1}) - f^*] \leq \theta\gamma_k\mu_k^{2\alpha-1}.$$

Hence, the induction statement holds for $k + 1$. We conclude that (13) holds for all $k \geq K$. As a consequence, (10) holds for all $k \geq K$ as well. To complete the proof, we need to show $\lim_{k\to\infty}\mathsf{E}[f(x_k)] = f^*$. This is an immediate result of (10) and Assumption 5(a). □

4. Iteratively regularized stochastic and deterministic LBFGS methods. In this section, our main goal is to develop an efficient update rule for matrix H_k of the (IR-SQN) scheme and establish a convergence rate result.

4.1. Background. Stochastic gradient methods are known to be sensitive to the choice of stepsizes. In our prior work [16, 31, 32], we address this challenge in part by developing self-tuned stepsizes under the strong convexity assumption. Another avenue to enhance the robustness of this scheme lies in incorporating curvature information of the objective function. A well-known updating rule for the matrix H_k that uses the curvature estimates is the BFGS update. The deterministic BFGS scheme, achieves a superlinear convergence rate (cf. Theorem 8.6 [23]) outperforming the deterministic gradient/subgradient method. In the classical deterministic BFGS scheme, the curvature information is incorporated within the algorithm using two terms: the first term is the displacement factor $s_k = x_{k+1} - x_k$, while the other is the change in the gradient mapping, $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, where ∇f denotes the gradient mapping of the deterministic objective function f. To have a well-defined update rule, it is essential that at each iteration, the curvature condition $s_k^T y_k > 0$ is satisfied. By maintaining this condition at each iteration, the positive definiteness of the approximate Hessian, denoted by B_k, is preserved. The BFGS update rule in deterministic regime also ensures that B_k satisfies a secant equation given by $B_{k+1}s_k = y_k$, which ensures that the approximate Hessian maps s_k into y_k.

To address optimization problems in the stochastic regime, a regularized BFGS update rule, namely (RES), was developed for problem (SO) under the strong convexity assumption [19]. In problems with a large dimension (see [3] for some examples), the implementation of this scheme becomes challenging. This is because the computation of B_k and its inverse become expensive. Moreover, at each iteration, a matrix of size $n \times n$ needs to be stored. To address these issues in large scale optimization problems, limited-memory variants of stochastic BFGS scheme, denoted by stochastic LBFGS, have been developed [3, 20]. The key idea in LBFGS update rule is that instead of storing the full $n \times n$ matrix at each iteration, a fixed number of vectors of size n are stored and used to update the approximate Hessian inverse.

4.2. Outline of the stochastic LBFGS scheme. The strong convexity property assumed in [3, 20] plays a key role in developing the LBFGS update rules and establishing the convergence. Note that in the absence of strong convexity, the curvature condition does not hold. To address this issue, a standard approach is to employ a damped variant of the BFGS update rule [23]. A drawback of this class of update rules is that there is no guarantee on the rate statements under such rules. Here, we resolve this issue through employing the properties of the regularized gradient map. This is carried out in (IR-SQN) by adding the regularization term $\mu_k(x_k - x_0)$ to the stochastic gradient mapping $\nabla F(x_k, \xi_k)$. To maintain the curvature condition, we consider updating the matrix H_k and the parameter μ_k in alternate steps. Keeping the regularization parameter constant in one iteration allows for maintaining the curvature condition. After updating H_k, in the subsequent iteration, we keep this matrix fixed and drop the value of the regularized parameter. Accordingly, the update rule for the regularization parameter μ_k is based on the following general procedure:

$$(15) \quad \begin{cases} \mu_k := \mu_{k-1}, & \text{if } k \text{ is odd,} \\ \mu_k < \mu_{k-1}, & \text{otherwise.} \end{cases}$$

Note that we allow for updating the stepsize sequence at each iteration. We construct the update rule in terms of the following two factors defined for any odd $k \geq 1$:

$$(16) \quad \begin{aligned} s_{\lceil k/2 \rceil} &:= x_k - x_{k-1}, \\ y_{\lceil k/2 \rceil} &:= \nabla F(x_k, \xi_{k-1}) - \nabla F(x_{k-1}, \xi_{k-1}) + \tau \mu_k^\delta s_{\lceil k/2 \rceil}, \end{aligned}$$

11

where $\tau > 0$ and $0 < \delta \le 1$ are parameters to control the level of regularization in the matrix H_k. Here, δ only controls the regularization for matrix H_k, but not that of the gradient direction. It is assumed that $\delta > 0$ to ensure that the perturbation term $\mu_k^\delta s_{\lceil k/2 \rceil} \to 0$, as $k \to \infty$. The update policy for H_k is defined as follows:

$$(17) \qquad H_k \triangleq \begin{cases} H_{k,m}, & \text{if } k \text{ is odd,} \\ H_{k-1}, & \text{otherwise,} \end{cases}$$

where $m < n$ (in the large scale settings, $m \ll n$) is the memory parameter and represents the number of pairs (s_i, y_i) to be stored to estimate H_k. Matrix $H_{k,m}$, for any odd $k \ge 2m - 1$, is updated using the following recursive formula:

$$(18) \qquad H_{k,j} := \left(\mathbf{I} - \frac{y_i s_i^T}{y_i^T s_i} \right)^T H_{k,j-1} \left(\mathbf{I} - \frac{y_i s_i^T}{y_i^T s_i} \right) + \frac{s_i s_i^T}{y_i^T s_i}, \quad 1 \le j \le m,$$

where $i \triangleq \lceil k/2 \rceil - (m - j)$ and we set $H_{k,0} := \frac{s_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil}}{y_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil}} \mathbf{I}$. Here, at odd iterations, matrix H_k is obtained recursively from $H_{k,0}, H_{k,1}, \ldots, H_{k,m-1}$. Note that computation of H_k at an odd k needs m pairs of (s_i, y_i). More precisely, H_k uses the following curvature information: $\{(s_i, y_i) \mid i = \lceil k/2 \rceil - m + 1, \lceil k/2 \rceil - m + 2, \ldots, \lceil k/2 \rceil \}$. For convenience, in the first $2m - 2$ iterations, we let H_k be the identity matrix. This allows for collecting the first set of m pairs (s_i, y_i), where $i = 1, 2, \ldots, m$, that is used at iteration $k := 2m - 1$ to obtain H_{2m-1}. The main differences between update rule (18) and that of the standard SQN schemes [3,20] are as follows: (i) The first distinction is with respect to the definition of y_i in (16). Here the term $\mu_k^\delta s_{\lceil k/2 \rceil}$ compensates for the lack of strong monotonicity of the gradient mapping and aids in establishing the curvature condition. (ii) Second, instead of obtaining the pair (s_i, y_i) at every iteration, we evaluate these terms only at odd iterations to allow for updating the regularization parameter satisfying (15).

Implementation of this stochastic LBFGS scheme requires computing the term $H_k \nabla F_k(x_k, \xi_k)$ at the kth iteration. This can be performed through a two-loop recursion with $\mathcal{O}(mn)$ number of operations (see Ch. 7, Pg. 178 in [23]). This will be shown for Algorithm 1 in Theorem 10(b).

In this section, we consider a stronger variant of Assumption 1 stated as follows:

ASSUMPTION 6.   (a) The function $F(x, \xi)$ is convex over $\mathbb{R}^n$ for any $\xi \in \Omega$.
(b) For any $\xi \in \Omega$, $F(\cdot, \xi)$ is continuously differentiable with Lipschitz continuous gradients over $\mathbb{R}^n$ with parameter $L_\xi > 0$. Moreover, $L := \sup_{\xi \in \Omega} L_\xi < \infty$.
(c) The optimal solution set X^* of problem (SO) is nonempty.

Next, in Lemma 7, we derive bounds on the eigenvalues of the matrix H_k and show that at iterations where H_k is updated, both the curvature condition and the secant equation hold. In the proof of Lemma 7, we will make use of the following result.

LEMMA 6. Let $0 < a_1 \le a_2 \le \ldots \le a_n$, and P and S be positive scalars such that $\sum_{i=1}^n a_i \le S$ and $\prod_{i=1}^n a_i \ge P$. Then, we have $a_1 \ge (n-1)! P / S^{n-1}$.

Proof. See Appendix 7.1.   □

LEMMA 7 (Properties of update rule (17)-(18)). Consider the (IR-SQN) method. Let H_k be given by the update rule (17)-(18), where s_i and y_i are defined in (16) and μ_k is updated according to the procedure (15). Let Assumption 6(a,b) hold. Then, the following results hold:

(a) *For any odd $k \geq 2m - 1$, the curvature condition holds, i.e., $s_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil} > 0$.*

(b) *For any odd $k \geq 2m - 1$, the secant equation holds, i.e., $H_k y_{\lceil k/2 \rceil} = s_{\lceil k/2 \rceil}$.*

(c) *For any $k \geq 2m - 1$, H_k satisfies Assumption 3 with the following values:*

$$\lambda_{\min} = \frac{1}{(m+n)\left(L + \tau \mu_0^\delta\right)}, \quad \lambda = \frac{(m+n)^{n+m-1}\left(L + \tau \mu_0^\delta\right)^{n+m-1}}{(n-1)!\tau^{(n+m)}},$$

(19) *and $\alpha = -\delta(n+m)$.*

More precisely, H_k is symmetric, $\mathsf{E}[H_k \mid \mathcal{F}_k] = H_k$ and

$$(20) \qquad \frac{1}{(m+n)\left(L + \tau \mu_0^\delta\right)}\mathbf{I} \preceq H_k \preceq \frac{(m+n)^{n+m-1}\left(L + \tau \mu_0^\delta\right)^{n+m-1}}{(n-1)!\left(\tau \mu_k^\delta\right)^{(n+m)}}\, \mathbf{I}.$$

Proof. See Appendix 7.2. □

In the following two lemmas, we provide update rules for the stepsize and the regularization parameter to ensure convergence in both an a.s. and mean sense for the proposed LBFGS scheme.

LEMMA 8 (Feasible tuning sequences for a.s. convergence (Proposition 4)). *Let the sequences γ_k and μ_k be given by the following rules:*

$$(21) \qquad\qquad \gamma_k = \frac{\gamma_0}{(k+1)^a}, \quad \mu_k = \frac{2^b \mu_0}{(k+\kappa)^b},$$

where $\kappa = 2$ if k is even and $\kappa = 1$ otherwise, γ_0 and μ_0 are positive scalars such that $\gamma_0 \mu_0 \leq L(m+n)$, and a, b, and $\delta \leq 1$ are positive scalars satisfying:

$$\frac{a}{b} > 1 + 2\delta(n+m), \quad a + b \leq 1, \quad a + 2b > 1, \quad \text{and} \quad a - \delta b(m+n) > 0.5.$$

Then, γ_k and μ_k satisfy Assumption 4 with $\lambda_{\min}$ and α in (19) and μ_k satisfies (15).

Proof. See Appendix 7.3. □

REMARK 1 (An example of feasible sequences). *The conditions on parameters a, b, γ_0, and μ_0 in Lemma 8 hold for $\gamma_0 = \mu_0 \leq \sqrt{L}$, $a = \frac{5}{6}$ and $b = \frac{1}{6}$, and $\delta = \frac{1}{m+n}$.*

LEMMA 9 (Feasible tuning sequences for convergence in mean (Proposition 5)). *Let the sequences γ_k and μ_k be given by (21), where γ_0 and μ_0 are positive scalars such that $\gamma_0 \mu_0 \leq L(m+n)$. Let, $0 < \delta \leq 1$ $a > 0$ and $b > 0$ satisfying:*

$$\frac{a}{b} > 1 + 2\delta(m+n), \quad a + b < 1, \quad \frac{a}{b} \leq 2\left(1 + \delta(m+n)\right).$$

Then, μ_k satisfies (15) and γ_k and μ_k satisfy Assumption 5 with any arbitrary $0 < \beta < 1$, $\rho \triangleq \gamma_0^{-1}\left(\mu_0 2^b\right)^{2+2\delta(m+n)}$, and with $\lambda_{\min}$ and α given by (19).

Proof. See Appendix 7.4. □

Algorithm 1 Iteratively Regularized Stochastic Limited-memory BFGS
___

1: **Input:** LBFGS memory parameter $m \geq 1$, Lipschitzian parameter $L > 0$, random initial point $x_0 \in \mathbb{R}^n$, initial stepsize $\gamma_0 > 0$, and initial regularization parameter $\mu_0 > 0$ such that $\gamma_0 \mu_0 \leq (m+n)L$, scalars $0 < \epsilon < \frac{1}{3}$, $\delta \in \left(0, \frac{1.5\epsilon}{n+m}\right)$, and $\tau > 0$;

2: Set $a := \frac{2}{3} - \epsilon + \frac{2\delta(n+m)}{3}$, $b := \frac{1}{3}$;

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

4:     Compute $\gamma_k := \frac{\gamma_0}{(k+1)^a}$ and $\mu_k := \frac{\mu_0 2^b}{(k+1+\mathrm{mod}(k+1,2))^b}$;

5:     Evaluate the stochastic gradient $\nabla F(x_k, \xi_k)$;

6:     **if** $\mathrm{mod}(k,2) = 1$ **then**

7:         Compute index $i := \lceil k/2 \rceil$;

8:         Compute vector $s_i := x_k - x_{k-1}$;

9:         Compute vector $y_i := \nabla F(x_k, \xi_{k-1}) - \nabla F(x_{k-1}, \xi_{k-1}) + \tau \mu_k^\delta s_i$;

10:         **if** $k > 2m$ **then**

11:             Discard the vector pair $\{s_{i-m}, y_{i-m}\}$ from storage;

12:         **end if**

13:     **end if**

14:     **if** $k < 2m - 1$ **then**

15:         Update solution iterate $x_{k+1} := x_k - \gamma_k \left(\nabla F(x_k, \xi_k) + \mu_k(x_k - x_0)\right)$;

16:     **else**

17:         Initialize Hessian inverse $H_{k,0} := \frac{s_i^T y_i}{y_i^T y_i} \mathbf{I}$;

18:         Initialize $q := \nabla F(x_k, \xi_k) + \mu_k(x_k - x_0)$;

19:         **for** $t = i : i - m + 1$ **do**

20:             Compute scalar $\alpha_{i-t+1} := \frac{s_t^T q}{s_t^T y_t}$;

21:             Update vector $q := q - \alpha_{i-t+1} y_t$;

22:         **end for**

23:         Initialize vector $r := H_{k,0} q$;

24:         **for** $t = i - m + 1 : i$ **do**

25:             Update vector $r := r + \left(\alpha_{i-t+1} - \frac{y_t^T r}{s_t^T y_t}\right) s_t$;

26:         **end for**

27:         Update solution iterate $x_{k+1} := x_k - \gamma_k r$;   ▷ LBFGS update

28:     **end if**

29: **end for**

                                        LBFGS two-loop recursion (brace spanning lines 17–27)
___

4.3. An efficient implementation with rate analysis. Algorithm 1 presents an efficient implementation of the proposed stochastic LBFGS scheme. Note that update rules for the stepsize and regularization parameter are specified in line #2 and #4. Before presenting the complexity analysis in Theorem 10, we make some comments on the choice of parameter τ.

REMARK 2. *As mentioned, the parameter $\tau > 0$ in line #9 in Algorithm 1 is used to control the level of the iterative regularization employed in the computation of matrix H_k. Intuitively, it may seem that a small choice for τ may reduce the distortion caused by the term $\mu_k^\delta s_i$ in approximating the Hessian inverse, and consequently, improve the performance of the algorithm. However, this may not be always the case. To see this, first note that the relation (20) shows the dependency of eigenvalues of H_k on the choice of τ. Recall that the relation (4) is a key assumption used in the convergence and rate analysis of the proposed method. It can be seen that when $\tau \to 0$, the right-*

14

hand side of (4) *will decrease to zero. This indicates that a small τ enforces a small value for the term $\gamma_0/\mu_0^{2\delta(n+m)}$. For example, assuming a fixed value for μ_0, this would lead to a small γ_0. This may have a negative impact on the performance of the algorithm. As such, it is not clear if a small τ can be always beneficial. A closer look into this trade off calls for a more detailed analysis of the finite-time performance of the algorithm, which is not the focus of our current work and remains as a future direction to our study.*

In Theorem 10(a), we establish the convergence rate of Algorithm 1. Moreover, in Theorem 10(b), we show that the term $H_k \nabla F_k(x_k, \xi_k)$ is computed efficiently using the LBFGS two-loop recursion in the algorithm with $\mathcal{O}(mn)$ complexity per iteration.

THEOREM 10 (Rate analysis for Algorithm 1). *Consider Algorithm 1. The following statements hold:*

(a) *Suppose Assumptions 2 and 6 hold. Then, there exists $K \geq 2m - 1$ such that*

$$\mathsf{E}[f(x_k)] - f^* \leq \left(\frac{\theta\gamma_0}{\left(\mu_0 \sqrt[3]{2}\right)^{1-2\alpha}} \right) \frac{1}{k^{\frac{1}{3}-\epsilon}}, \quad \textit{for all } k > K,$$

where θ is given by (11), *and $\lambda_{\min}$, λ, and α are given by* (19).

(b) *Let the scheme be at the kth iteration where $k \geq 2m - 1$. Then, by the end of the LBFGS two-loop recursion, i.e., line #26 in Algorithm 1, we have*

(22) $$r = H_k \left(\nabla F(x_k, \xi_k) + \mu_k(x_k - x_0) \right),$$

where H_k is defined by (17).

Proof. (a) First, we show that the conditions of Proposition 5 are satisfied. Assumption 1 holds as a consequence of Assumption 6. From Lemma 7(c), Assumption 3 holds for any $k \geq 2m - 1$ as well. To show that Assumption 5 holds, we apply Lemma 9. We have

$$\frac{a}{b} = \frac{\frac{2}{3} - \epsilon + \frac{2\delta(n+m)}{3}}{1/3} = 2 - 3\epsilon + 2\delta(n+m) > 1 + 2\delta(m+n),$$

where we used $\epsilon < \frac{1}{3}$. Moreover, since $\delta < \frac{1.5\epsilon}{n+m}$, we have $a + b = 1 - \epsilon + \frac{2\delta(n+m)}{3} < 1$. Also, from the values of a and b we have $2b(1 + \delta(m+n)) = a + \epsilon > a$. Thus, the conditions of Lemma 9 hold. This implies that there exists $K_0 > 0$ such that for any $k \geq K_0$, the sequences γ_k and μ_k satisfy Assumption 5 with any arbitrary $0 < \beta < 1$ and for $\rho = \gamma_0^{-1} \left(\mu_0 2^b\right)^{2+2\delta(m+n)}$, and with $\lambda_{\min}$, λ and α given by (19). Let us define $K := \max\{K_0, 2m-1\}$. Since all conditions in Proposition 5 are met, from (10), (21), and by substituting values of a, b, and α, for any $k \geq K$ we obtain

$$\mathsf{E}[f(x_{k+1})] - f^* \leq \theta\gamma_{k-1}\mu_{k-1}^{2\alpha-1} = \frac{\theta\gamma_0(k+\kappa-1)^{(1-2\alpha)/3}}{\left(\mu_0\sqrt[3]{2}\right)^{1-2\alpha} k^{2/3-\epsilon-\frac{2}{3}\alpha}}$$

$$\leq \frac{\theta\gamma_0(k+1)^{(1-2\alpha)/3}}{\left(\mu_0\sqrt[3]{2}\right)^{1-2\alpha}(k+1)^{2/3-\epsilon-\frac{2}{3}\alpha}} = \left(\frac{\theta\gamma_0}{\left(\mu_0\sqrt[3]{2}\right)^{1-2\alpha}} \right) \frac{1}{(k+1)^{\frac{1}{3}-\epsilon}}.$$

Through a change of variable from $k + 1$ to k, we conclude the result.

(b) To show (22), it suffices to show that

$$r = \begin{cases} H_{k,m}\left(\nabla F(x_k, \xi_k) + \mu_k(x_k - x_0)\right), & \text{if } k \text{ is odd,} \\ H_{k-1,m}\left(\nabla F(x_k, \xi_k) + \mu_k(x_k - x_0)\right), & \text{otherwise,} \end{cases}$$

where $H_{k,m}$ is defined by the recursion (18) for an odd k. First, consider the case that $k \geq 2m - 1$ is an odd number. As such, at the kth iteration, from line #7, we have $i := \lceil k/2 \rceil$. For clarity of the presentation, throughout this proof, we use K (instead of i), i.e., $K \triangleq \lceil k/2 \rceil$ and q_{K-t+1} is used to denote the value of the vector $q \in \mathbb{R}^n$ after being updated at iteration t in line #21. Similarly, we use r_{t-K+m} to denote the value of the vector $r \in \mathbb{R}^n$ after being updated at iteration t in line #25. Also, we use the following definitions:

$$q_0 \triangleq \nabla F(x_k, \xi_k) + \mu_k(x_k - x_0), \quad r_0 \triangleq H_{k,0}q_m,$$

$$\rho_j \triangleq \frac{1}{y_j^T s_j}, \quad \text{and } V_j \triangleq \mathbf{I} - \rho_j y_j s_j^T, \quad \text{for all } j = K - (m-1), \ldots, K.$$

Consider relation (18). By applying this recursive relation repeatedly, we obtain

$$\begin{aligned}
(23) \quad H_{k,m} = &\left(\prod_{j=1}^{m} V_{K-(m-j)}\right)^T H_{k,0} \left(\prod_{j=1}^{m} V_{K-(m-j)}\right) \\
& + \rho_{K-m+1} \left(\prod_{j=2}^{m} V_{K-(m-j)}\right)^T s_{K-m+1} s_{K-m+1}^T \left(\prod_{j=2}^{m} V_{K-(m-j)}\right) \\
& + \rho_{K-m+2} \left(\prod_{j=3}^{m} V_{K-(m-j)}\right)^T s_{K-m+2} s_{K-m+2}^T \left(\prod_{j=3}^{m} V_{K-(m-j)}\right) \\
& + \ldots \\
& + \rho_{K-1} V_K^T s_{K-1} s_{K-1}^T V_K \\
& + \rho_K s_K s_K^T.
\end{aligned}$$

Next, we derive a formula for q_t. From lines #20-21 in the algorithm, we have

$$\begin{aligned}
q_{K-t+1} = q_{K-t} - \alpha_{K-t+1} y_t &= q_{K-t} - \rho_t \left(s_t^T q_{K-t}\right) y_t = q_{K-t} - \rho_t \left(y_t s_t^T\right) q_{K-t} \\
&= \left(\mathbf{I} - \rho_t y_t s_t^T\right) q_{K-t} = V_t q_{K-t}, \quad \text{for all } t = K, K-1, \ldots, K-m+1.
\end{aligned}$$

From the preceding relation, we obtain

$$(24) \quad q_\ell = \left(\prod_{j=m-\ell+1}^{m} V_{K-(m-j)}\right) q_0, \quad \text{for all } \ell = 1, 2, \ldots, m.$$

From the update rule for α_{i-t+1} in line #20, using the definition of ρ_t, and applying the previous relation, we have $\alpha_1 = \rho_K s_K^T q_0$ and

$$(25) \quad \alpha_\ell = \rho_{K-\ell+1} s_{K-\ell+1}^T \left(\prod_{j=m-\ell+2}^{m} V_{K-(m-j)}\right) q_0, \quad \text{for all } \ell = 2, 3, \ldots, m.$$

16

Multiplying both sides of (23) by q_0 and employing (24) and (25), we obtain

$$
(26) \qquad H_{k,m}q_0 = \left(\prod_{j=1}^{m} V_{K-(m-j)} \right)^T H_{k,0}q_m + \left(\prod_{j=2}^{m} V_{K-(m-j)} \right)^T s_{K-m+1}\alpha_m
$$

$$
+ \left(\prod_{j=3}^{m} V_{K-(m-j)} \right)^T s_{K-m+2}\alpha_{m-1} + \ldots + V_K^T s_{K-1}\alpha_2 + s_K\alpha_1.
$$

Next, we derive a formula for r_t. From line #25 in the algorithm, we have

$$
\begin{aligned}
r_{t-K+m} &= r_{t-K+m-1} + \left(\alpha_{K-t+1} - \rho_t y_t^T r_{t-K+m-1} \right) s_t \\
&= r_{t-K+m-1} - \rho_t s_t y_t^T r_{t-K+m-1} + \alpha_{K-t+1} s_t \\
&= V_t^T r_{t-K+m-1} + \alpha_{K-t+1} s_t, \quad \text{for all } t = K-m+1, \ldots, K-1, K.
\end{aligned}
$$

Combining the preceding two relations, we obtain

$$
r_\ell = V_{K-(m-\ell)}^T r_{\ell-1} + \alpha_{m-\ell+1} s_{K-(m-\ell)}, \quad \text{for all } \ell = 1, 2, \ldots, m.
$$

Using the preceding equation repeatedly, we obtain

$$
(27) \qquad r_m = \left(\prod_{j=1}^{m} V_{K-(m-j)} \right)^T r_0 + \alpha_m \left(\prod_{j=2}^{m} V_{K-(m-j)} \right)^T s_{K-m+1}
$$

$$
+ \alpha_{m-1} \left(\prod_{j=3}^{m} V_{K-(m-j)} \right)^T s_{K-m+2} + \ldots + \alpha_2 V_K^T s_{K-1} + \alpha_1 s_K.
$$

From (27) and (26), and the definition of r_0, we obtain $r_m = H_{k,m}q_0$. Taking to account the definition of q_0, the desired result holds for any odd $k \geq 2m-1$. Now, consider the case where $k \geq 2m-1$ is an even number. This implies that the "if" condition in line #6 is skipped and as such, the value of i is not updated from the iteration $k-1$, i.e., $i = \lceil (k-1)/2 \rceil$. Consequently, the LBFGS two-loop recursion at an even k uses the following pairs

$$
\{ (s_\ell, y_\ell) \mid \ell = \lceil (k-1)/2 \rceil - m + 1, \lceil (k-1)/2 \rceil - m + 2, \ldots, \lceil (k-1)/2 \rceil \}.
$$

Now, considering the definition (18) for $k-1$, the desired relation can be shown following the same steps discussed for the case the iteration number is odd. $\square$

4.4. Analysis of the deterministic case. Our goal in the remainder of this section lies in establishing the convergence and rate statement for the deterministic LBFGS scheme. Consider the following regularized deterministic LBFGS method:

$$
\text{(IR-LBFGS)} \qquad x_{k+1} := x_k - \gamma_k H_k \left(\nabla f(x_k) + \mu_k (x_k - x_0) \right), \quad \text{for all } k \geq 0,
$$

where H_k is given by the update rule (17), μ_k is updated according to (15), and for an odd $k \geq 1$ we set

$$
\begin{aligned}
s_{\lceil k/2 \rceil} &:= x_k - x_{k-1}, \\
(28) \qquad y_{\lceil k/2 \rceil} &:= \nabla f(x_k) - \nabla f(x_{k-1}) + \tau \mu_k^\delta s_{\lceil k/2 \rceil}.
\end{aligned}
$$

THEOREM 11 (Convergence and rate analysis of iteratively regularized deterministic LBFGS method). *Let x_k be generated by the* IR-LBFGS *method. Suppose Assumption 1 holds. Let $\lambda_{\min}$, λ and α be given by (19). Then the following hold.*
(a) Let μ_k satisfies (15). If γ_k and μ_k satisfy the following relation:

$$(29) \qquad \gamma_k \mu_k^{2\alpha} \leq \frac{\lambda_{\min}}{\lambda^2(L+\mu_0)}, \quad \text{for all } k \geq 0,$$

then, for any $k \geq 0$, we have

$$(30) \qquad f_{k+1}(x_{k+1}) - f^* \leq (1 - \lambda_{\min}\mu_k\gamma_k)(f_k(x_k) - f^*) + \frac{\lambda_{\min}\, dist^2(x_0, X^*)}{2}\mu_k^2\gamma_k.$$

(b) Let γ_k and μ_k be given by the update rule (21) where $a, b > 0$ and $0 < \delta \leq 1$ satisfy

$$\frac{a}{b} > 2\delta(n+m) \quad a + b \leq 1, \quad a + 2b > 1.$$

Then, $\lim_{k\to\infty} f(x_k) = f^*$. Specifically, for $a = \frac{4}{5}$, $b = \frac{1}{5}$, and $\delta = \frac{1}{m+n}$, this result holds.
(c) Let $\epsilon \in (0,1)$ be an arbitrary small scalar. Let γ_k and μ_k be given by the update rule (21) where $a = \epsilon$, $b = 1 - \epsilon$. Also, assume $\delta \in \left(0, \frac{\epsilon}{2(n+m)(1-\epsilon)}\right)$. Let γ_0 and μ_0 satisfy the following condition:

$$(31) \qquad \gamma_0\mu_0 \geq (n+m)\left(L + \tau\mu_0^\delta\right).$$

Then, there exists K such that

$$(32) \qquad f(x_k) - f^* \leq \frac{\Gamma}{(k+1)^{1-\epsilon}}, \quad \text{for all } k \geq K,$$

where $\Gamma \triangleq \max\left\{(K+1)^{1-\epsilon}\left(f_K(x_K) - f^*\right), \frac{\lambda_{\min}\gamma_0\mu_0^2\, dist^2(x_0, X^*)}{4^a(\lambda_{\min}\gamma_0\mu_0 - b)}\right\}$.

Proof. (a) The conditions of Lemma 7 are met indicating that Assumption 7 holds. Assumption 2 is clearly met with $\nu = 0$ as the problem is deterministic. Therefore, all of the conditions of Lemma 3 are satisfied and thus (4) holds. Substituting $\nu = 0$ in (4) and eliminating the expectation operator yields the desired inequality.
(b) First, we show that (30) holds. We can write

$$\gamma_k\mu_k^{2\alpha} = \frac{\gamma_0}{(2^b\mu_0)^{-2\alpha}}(k+1)^{-a}(k+\kappa)^{-2\alpha b} \leq \frac{\gamma_0}{(2^b\mu_0)^{-2\alpha}}(k+1)^{-a-2\alpha b}.$$

Note that the assumption that $a > 2b\delta(n+m)$, implies that $-a - 2\alpha b < 0$. Therefore, $\gamma_k\mu_k^{2\alpha} \to 0$ showing that there exists K_0 such that for any $k \geq K_0$, (30) holds. We apply Lemma 2 to the inequality (30) by setting

$$\alpha_k := \lambda_{\min}\gamma_0\mu_0, \quad \beta_k := \frac{\lambda_{\min}dist^2(x_0, X^*)}{2}\mu_k^2\gamma_k, \quad v_k := f_k(x_k) - f^*.$$

From $a + b \leq 1$, we have $\sum_{k=0}^\infty \alpha_k = \infty$. Also, $a + 2b > 1$ indicates that $\sum_{k=0}^\infty \beta_k < \infty$. Since all conditions of Lemma 2 are met, we have $f_k(x_k) \to f^*$. Recalling Definition 1, this implies that $f(x_k) \to f^*$.

(c) First, we show that by the given update rules for γ_k and μ_k, relation (30) holds. Note that $\alpha = -\delta(n+m)$. Therefore, we can write

$$\gamma_k \mu_k^{2\alpha} = \frac{\gamma_0(k+\kappa)^{2(m+n)\delta b}}{(\mu_0 2^b)^{2(m+n)\delta}(k+1)^a} \leq \frac{\gamma_0(k+2)^{2(m+n)\delta b}}{(\mu_0 2^b)^{2(m+n)\delta}(k+1)^a}$$

(33)
$$= \frac{\gamma_0(1+\frac{1}{k+1})^{2(m+n)\delta b}}{(\mu_0 2^b)^{2(m+n)\delta}(k+1)^{a-2(m+n)\delta b}}.$$

Using the condition on δ, we have $a - 2(m+n)b\delta = \epsilon - 2(1-\epsilon)\delta(m+n) > 0$. Thus, relation (33) indicates that there exists K_1 such that for any $k \geq K_1$, (30) holds. Besides, since a and b are positive, there exits K_2 such that for any $k \geq K_2$, we have $(1 - \lambda_{\min}\gamma_k\mu_k) > 0$. Let us now define $K := \max\{K_1, K_2, 2m-1\}$. Next, we use induction on k to show (32). For $k = K$, it clearly holds. Let us assume (32) holds for $k > K$. Let e_k denote $f_k(x_k) - f^*$. From (30) and the update rules of γ_k and μ_k we can write

$$e_k \leq \left(1 - \frac{\lambda_{\min}\gamma_0\mu_0 2^b}{k^a(k+\kappa-1)^b}\right)e_{k-1} + \frac{\lambda_{\min}\text{dist}^2(x_0, X^*)\gamma_0\mu_0^2 2^{2b-1}}{k^a(k+\kappa-1)^{2b}}$$

$$\leq \left(1 - \frac{\lambda_{\min}\gamma_0\mu_0 2^b}{k^a(k+1)^b}\right)e_{k-1} + \frac{\lambda_{\min}\text{dist}^2(x_0, X^*)\gamma_0\mu_0^2 2^{2b-1}}{k^{a+2b}}$$

(34)
$$\leq \left(1 - \frac{\lambda_{\min}\gamma_0\mu_0}{k}\right)e_{k-1} + \frac{\lambda_{\min}\text{dist}^2(x_0, X^*)\gamma_0\mu_0^2 2^{2b-1}}{k^{a+2b}},$$

where κ is defined in (21), and the last inequality is implied by $\frac{k^a(k+1)^b}{k^{a+b}} \leq 2^b$ for $k \geq 1$. Note that since $k \geq K_2$, the term $\left(1 - \frac{\lambda_{\min}\gamma_0\mu_0}{k^{a+b}}\right)$ in (34) is nonnegative. Therefore, we can replace e_{k-1} by its upper bound $\frac{\Gamma}{k^b}$ in (34). Doing so and noticing that $a + b = 1$, we obtain

(35)
$$e_k \leq \left(1 - \frac{C_1}{k}\right)\frac{\Gamma}{k^b} + \frac{C_2}{k^{b+1}},$$

where we define $C_1 \triangleq \lambda_{\min}\gamma_0\mu_0$ and $C_2 \triangleq \lambda_{\min}\text{dist}^2(x_0, X^*)\gamma_0\mu_0^2 2^{2b-1}$. Using (35), to show that $e_k \leq \frac{\Gamma}{(k+1)^{1-a}}$, it is enough to show that

$$\Gamma\left(\frac{1}{k^b} - \frac{1}{(k+1)^b}\right) \leq \frac{C_1\Gamma - C_2}{k^{b+1}}.$$

Rearranging the terms, we need to verify that $\Gamma \geq \frac{C_2}{C_1-C_3}$ and $C_3 < C_1$, where C_3 is an upper bound on $\sup_{k\geq 1}\left\{k^{b+1}\left(\frac{1}{k^b} - \frac{1}{(k+1)^b}\right)\right\}$. We claim that $C_3 := b$ is a feasible choice. To prove this, we need to show that $k^{b+1}\left(\frac{1}{k^b} - \frac{1}{(k+1)^b}\right) \leq b$, or equivalently,

$$\left(1 - \frac{1}{k+1}\right)^b \geq 1 - \frac{b}{k}, \quad \text{for all } k \geq 1.$$

Consider the function $g(x) := (1 - \frac{1}{1+x})^b + \frac{b}{x} - 1$ for $x \geq 1$. we have

$$g'(x) = \frac{b}{(1+x)^2}\left(1 - \frac{1}{1+x}\right)^{1-b} - \frac{b}{x^2} = \frac{b}{(1+x)^2}\left(\left(\frac{x+1}{x}\right)^{1-b} - \left(\frac{x+1}{x}\right)^2\right) \leq 0,$$

19

due to $0 < b < 1$. Hence, g is non-increasing implying that it suffices to show $g(1) \geq 0$, i.e., $2^b(1-b) \leq 1$. Let us define $h(x) := 2^x(1-x)$ for $0 < x < 1$. We have $h'(x) = 2^x(\ln(2)(1-x) - 1)$. This indicates that $h'(x) < 0$ over $x \in (0,1)$, implying that $h(b) \leq h(0) = 1$. Hence, we conclude that $C_3 := b$ is a feasible choice. To show that $C_3 < C_1$ holds, we need to verify that $C_1 > b$. This is true due to (31). To complete the proof we need to show $\Gamma \geq \frac{C_2}{C_1 - b}$. This holds from the definition of Γ. □

5. Numerical experiments. In this section, we present the implementation results of Algorithm 1 on a classification application. The Reuters Corpus Volume I (RCV1) data set [11] is a collection of news-wire stories produced by Reuters. After the tokenization process, each article is converted to a sparse binary vector, in that 1 denotes the existence and 0 denotes nonexistence of a token in the corresponding article. We consider a subset of the data with $N = 100,000$ articles and $n = 138,921$ tokens. The articles are categorized into different hierarchical groups. Here we focus our attention on the binary classification of the articles with respect to the Markets class. We consider the logistic regression loss minimization problem given as follows:

$$\text{(LRM)} \qquad \min_{x \in \mathbb{R}^n} f(x) := \frac{1}{N} \sum_{i=1}^{N} \ln\left(1 + \exp\left(-u_i^T x v_i\right)\right),$$

where $u_i \in \mathbb{R}^n$ is the input binary vector associated with article i and $v_i \in \{-1, 1\}$ denotes the class of the ith article. We run three experiments. Of these, in Section 5.1, we compare the performance of Algorithm 1 (with $\tau = 1$) with that of standard SQN methods. In Sections 5.2 and 5.3, we provide comparisons of Algorithm 1 with SAGA [5] and with IAG [9] applied to regularized problems, respectively.

5.1. Comparison with standard SQN schemes. To solve problem (LRM), the standard LBFGS methods in [3, 20] solve an approximate problem of the form

$$\text{(Regularized LRM)} \qquad \min_{x \in \mathbb{R}^n} f(x) := \frac{1}{N} \sum_{i=1}^{N} \ln\left(1 + \exp\left(-u_i^T x v_i\right)\right) + \frac{\eta}{2}\|x\|^2,$$

where $\eta > 0$ is an arbitrary regularization parameter. To perform the first experiment, we consider comparison of Algorithm 1 with three variants of the standard LBFGS schemes, all denoted by RS-LBFGS (see Figure 1). In RS-LBFGS schemes, we use the stepsize of the form $\gamma_k = \frac{\gamma_0}{k+1}$ and drop η at epochs of 400 iterations using a decay factor denoted by $\rho \in (0, 1]$. Of these, in the first scheme, we assume $\rho = 1$, meaning that η is kept constant throughout the implementation of the SQN scheme. In the second scheme, we use $\rho = 0.5$. This means for example, after every 400 iterations, we set $\eta := 0.5\eta$. In the third scheme, we use $\rho = 0.3$. We let $\gamma_0 \in \{10, 0.5, 0.1\}$, $\eta_0 \in \{1, 0.5, 0.01\}$, $m \in \{2, 5\}$, $N = 10^4$, and x_0 be the origin. In all cases, we use five sample paths to calculate the average value of the objective function in (LRM).

Insights: We observe that Algorithm 1 performs uniformly better than the three variants of the standard SQN scheme under different tuning rules for the regularization parameter. This suggests that for merely convex stochastic optimization, SQN schemes using the tuning rules for the stepsize and regularization parameter given as $\gamma_k \approx 1/\sqrt[3]{k^2}$ and $\mu_k \approx 1/\sqrt[3]{k}$ have a faster convergence speed.

5.2. Comparison with SAGA on merely convex problems. Recall that in addressing the finite-sum minimization problems with merely convex objectives, employing averaging and under a constant stepsize, SAGA admits a sublinear convergence rate of $\mathcal{O}\left(\frac{N}{k}\right)$ [5]. The simulation results are provided in Figure 2. These
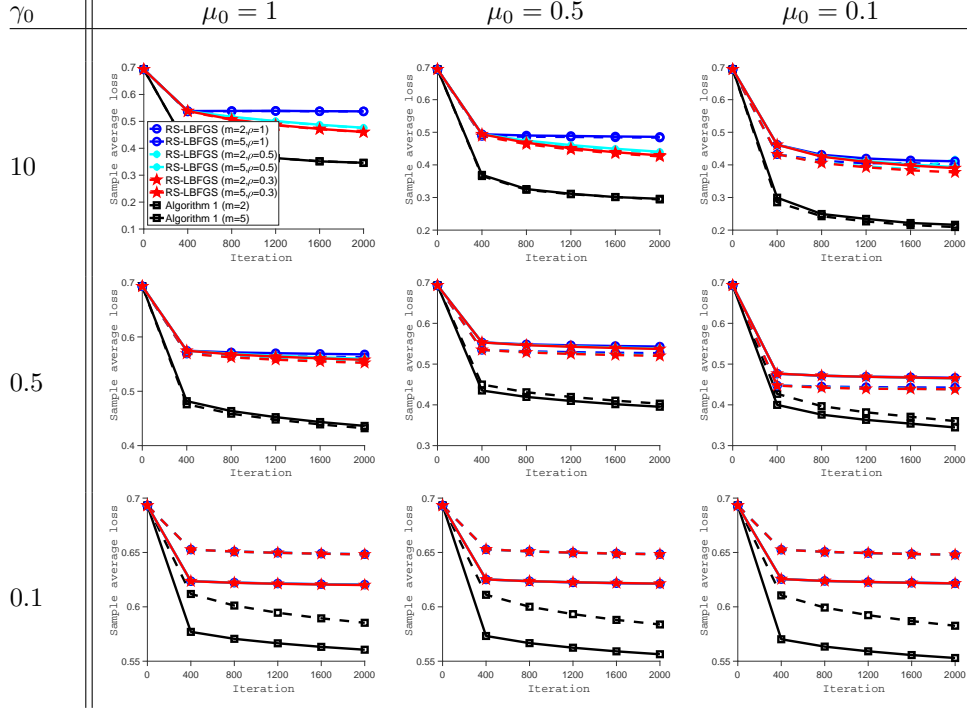
Fig. 1: Algorithm 1 vs. stochastic LBFGS under constant regularization (i.e., $\rho = 1$), and under piece-wise constant regularization (i.e., $\rho = 0.5, 0.3$), where ρ is the decay ratio of the regularization parameter at each epoch of 400 iterations.

results include different sample sizes, i.e., $N \in \{10^3, 10^4, 10^5\}$, different initial conditions for SAGA, and different choices of the stepsizes and the initial regularization parameter for Algorithm 1. SAGA uses the evaluation of the gradient map of the component functions at the starting point. Here, we use three different values for the evaluated gradient maps at the starting point, i.e., the origin, to study the sensitivity of SAGA with respect to the initial conditions. Of these, in initial condition 3, we use the exact value of the gradient maps, while in initial condition 2, we perturb values of the gradient maps. This perturbation is increased in initial condition 1.

Insights: From Figure 2, we observe that Algorithm 1 competes well with SAGA. We discuss the comparisons as follows: (i) A computational burden in implementation of SAGA is the memory requirement of this scheme. Generally speaking, SAGA requires storing a matrix of $\mathcal{O}(Nn)$ at each iteration. Exceptions include the case where the objective function is in terms of a linear regression model function (e.g., in (LRM)). This is in contrast with Algorithm 1 where the memory requirement is $\mathcal{O}(mn)$. (ii) As expected, the performance of SAGA deteriorates when the sample size increases. However, the performance of Algorithm 1 seems to be more robust with respect to the increase in the sample size. (iii) The performance of SAGA seems to be moderately sensitive to the initial conditions.

5.3. Comparison with IAG. Recall that in solving finite-sum minimization problems with μ-strongly convex objectives, using a constant stepsize, (non-averaging) IAG admits a linear convergence rate of $\mathcal{O}\left(\left(1 - (\mu/N)^2\right)^{2k}\right)$ where N is the number
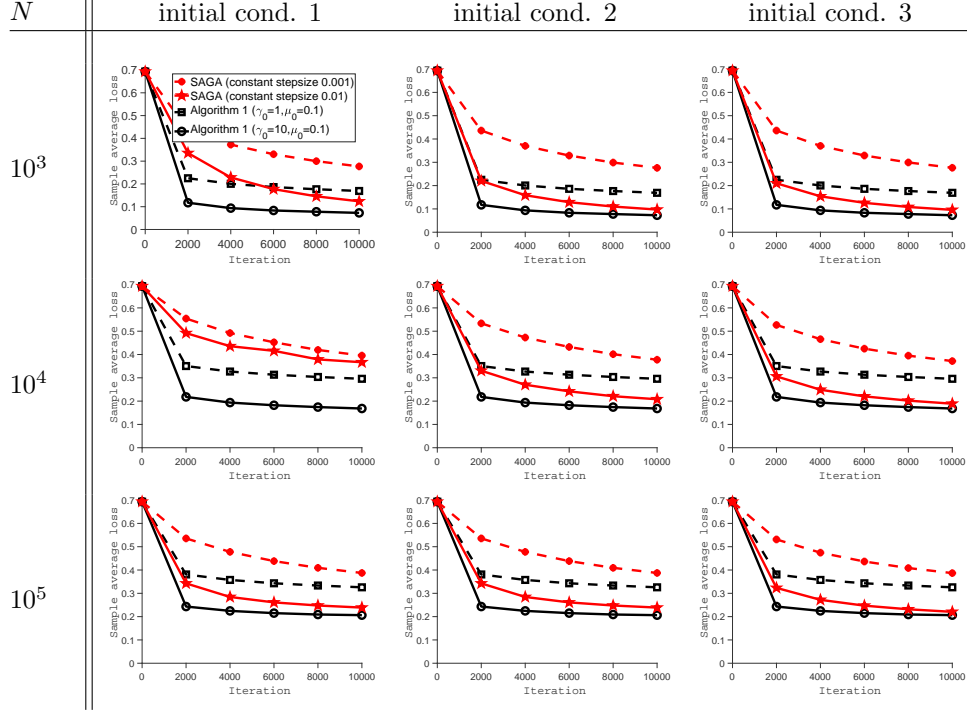
Fig. 2: Algorithm 1 vs. SAGA (with averaging) with different choices of constant stepsize, different sample sizes, and different initial value of the gradient of component functions.

of component functions (cf. [9]). Accordingly, to do the numerical comparisons with IAG, we regularize problem (LRM) with a constant $\mu_{IAG} > 0$. Figure 3 shows the simulation results for different choices of μ_{IAG}, N, IAG stepsize, and the initial stepzie and regularization parameter of Algorithm 1.

Insights: (i) Due to the excessive memory requirements of $\mathcal{O}(nN)$ associated with IAG, such a scheme becomes challenging to implement when n becomes large as in this case where $n = 138,921$. Consequently, we use a sample size $N \in \{1000, 2000, 5000\}$. However, Algorithm 1 only requires memory of $\mathcal{O}(nm)$, allowing for implementations with large values of N. (ii) Similar to SAGA, the performance of IAG is deteriorated when the sample size increases. However, the performance of Algorithm 1 seems to be more robust with changes in the sample size. (iii) For each fixed value of N, despite the change in the value of μ_{IAG}, the performance of IAG in terms of the true objective function in (LRM) does not necessarily improve. Importantly, this observation suggests that in the standard regularization approach, tuning the regularization parameter could be computationally expensive.

6. Concluding remarks. We consider stochastic quasi-Newton (SQN) methods for solving large scale stochastic optimization problems with smooth but unbounded gradients. Much of the past research on convergence rates of these algorithms relies on the strong convexity of the objective function. We employ an iterative regularization scheme where the regularization parameter is updated iteratively within the algorithm. We establish the convergence in an a.s. sense and a mean sense. Moreover, we prove that the iterates generated by the iteratively regularized stochastic LBFGS
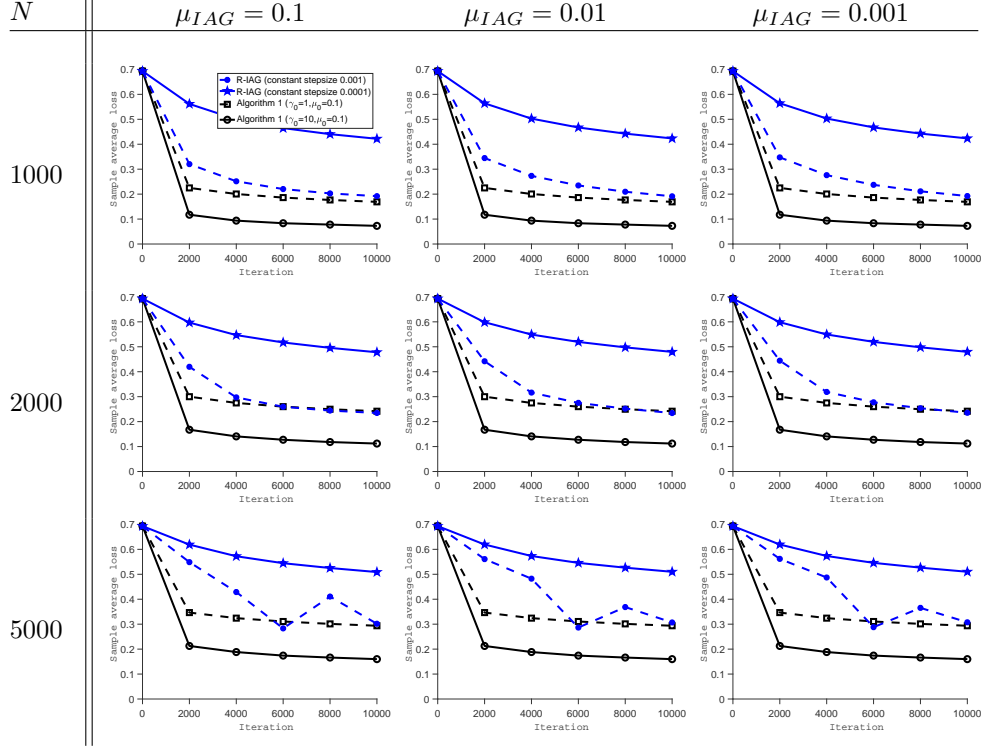
Fig. 3: Algorithm 1 vs. regularized IAG (non-averaging) with different choices of the regularization parameter, different choices of the stepsize, and different sample sizes.

scheme converges to an optimal solution at the rate $\mathcal{O}\left(\frac{1}{k^{1/3-\epsilon}}\right)$ for arbitrary small $\epsilon > 0$. The deterministic variant of this algorithm achieves the rate $\mathcal{O}\left(\frac{1}{k^{1-\epsilon}}\right)$. The numerical experiments performed on a large scale classification problem indicate that the proposed LBFGS scheme performs well compared to methods such as standard SQN schemes, and other first-order schemes such as SAGA and IAG.

7. Appendix.

7.1. Proof of Lemma 6. From $0 < a_1 \leq \ldots \leq a_n$, we can write

$$(n - (i-1))a_i \leq \sum_{j=1}^{n} a_j, \qquad \text{for all } i \in \{1, \ldots, n\}.$$

Invoking $\sum_{i=1}^{n} a_i \leq S$, we obtain $a_i \leq \frac{S}{n-(i-1)}$, for all $i \in \{1, \ldots, n\}$. From the preceding relation and that $\prod_{j=1}^{n} a_j \geq P$, we can obtain $a_1 \geq (n-1)! P / S^{n-1}$.

7.2. Proof of Lemma 7.

Proof. Throughout, we let $\lambda_{k,\min}$, $\lambda_{k,\max}$, and B_k denote the minimum eigenvalue, maximum eigenvalue, and inverse of matrix H_k in (18), respectively. It can be seen, by induction on k, that H_k is symmetric and $\mathcal{F}_k$ measurable. We use induction on odd values of $k \geq 2m - 1$ to show that parts (a), (b), and (c) hold. Suppose $k \geq 2m - 1$ is odd and for any odd $t < k$, we have $s_{\lceil t/2 \rceil}^T y_{\lceil t/2 \rceil} > 0$, $H_t y_{\lceil t/2 \rceil} = s_{\lceil t/2 \rceil}$, and (20)

for t. We show that these statements also hold for k as well. First, we show that the curvature condition holds. We can write

$$
\begin{aligned}
s_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil} &= (x_k - x_{k-1})^T (\nabla F(x_k, \xi_{k-1}) - \nabla F(x_{k-1}, \xi_{k-1}) + \tau \mu_k^\delta (x_k - x_{k-1})) \\
&= (x_k - x_{k-1})^T (\nabla F(x_k, \xi_{k-1}) - \nabla F(x_{k-1}, \xi_{k-1})) + \tau \mu_k^\delta \|x_k - x_{k-1}\|^2 \\
&\geq \tau \mu_k^\delta \|x_k - x_{k-1}\|^2,
\end{aligned}
$$

where the inequality follows from the monotonicity of the gradient map $\nabla F(\cdot, \xi)$. Next, we show that $\|x_k - x_{k-1}\|^2 > 0$. From the induction hypothesis and that $k - 2$ is odd, H_{k-2} is positive definite. Moreover, from the update rule (17) and that $k - 2$ is odd, we have $H_{k-1} = H_{k-2}$. Therefore, H_{k-1} is also positive definite. Without loss of generality, we assume $\nabla F(x_{k-1}, \xi_{k-1}) + \mu_{k-1}(x_{k-1} - x_0) \neq 0$[1]. Since H_{k-1} is positive definite, we have

$$
H_{k-1} \left(\nabla F(x_{k-1}, \xi_{k-1}) + \mu_{k-1}(x_{k-1} - x_0) \right) \neq 0,
$$

implying that $x_k \neq x_{k-1}$. Hence $s_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil} \geq \tau \mu_k^\delta \|x_k - x_{k-1}\|^2 > 0$, where the second inequality is a consequence of $\tau, \mu_k > 0$. Thus, the curvature condition holds. Next, we show that (20) holds for k. It is well-known that using the Sherman-Morrison-Woodbury formula, B_k is equal to $B_{k,m}$ given by

(36)

$$
B_{k,j} = B_{k,j-1} - \frac{B_{k,j-1} s_i s_i^T B_{k,j-1}}{s_i^T B_{k,j-1} s_i} + \frac{y_i y_i^T}{y_i^T s_i}, \quad i := \lceil k/2 \rceil - (m - j), \quad 1 \leq j \leq m,
$$

where s_i and y_i are defined by (16) and $B_{k,0} = \frac{y_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil}}{s_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil}} \mathbf{I}$. Note that with j varying between 1 to m, the index i takes values in $\{\lceil k/2 \rceil - m + 1, \lceil k/2 \rceil - m + 2, \ldots, \lceil k/2 \rceil\}$. First, we show that for any i in this range,

(37)
$$
\tau \mu_k^\delta \leq \frac{\|y_i\|^2}{y_i^T s_i} \leq L + \tau \mu_k^\delta,
$$

where L is the Lipschitzian parameter of the gradient mapping ∇F given by Assumption 6(b). Let us define the function $h(x) \triangleq F(x, \xi_{i-1}) + \tau \frac{\mu_k^\delta}{2} \|x - x_0\|^2$ for fixed i and k. Note that this function is strongly convex and has a gradient mapping of the form $\nabla F + \tau \mu_k^\delta (\mathbf{I} - x_0)$ that is Lipschitz with parameter $L + \tau \mu_k^\delta$. For a convex function h with Lipschitz gradient with parameter $L + \tau \mu_k^\delta$, the following inequality, referred to as co-coercivity property, holds for any $x_1, x_2 \in \mathbb{R}^n$ (see [24], Pg. 24 , Lemma 2):

$$
\|\nabla h(x_2) - \nabla h(x_1)\|^2 \leq \left(L + \tau \mu_k^\delta \right) (x_2 - x_1)^T (\nabla h(x_2) - \nabla h(x_1)).
$$

Substituting x_2 by x_i, x_1 by x_{i-1}, and recalling (16), the preceding inequality yields

(38)
$$
\|y_i\|^2 \leq \left(L + \tau \mu_k^\delta \right) s_i^T y_i.
$$

Note that function h is strongly convex with parameter $\tau \mu_k^\delta$. Applying the Cauchy-Schwarz inequality, we can write

$$
\frac{\|y_i\|^2}{s_i^T y_i} \geq \frac{\|y_i\|^2}{\|s_i\| \|y_i\|} = \frac{\|y_i\|}{\|s_i\|} \geq \frac{\|y_i\| \|s_i\|}{\|s_i\|^2} \geq \frac{y_i^T s_i}{\|s_i\|^2} \geq \tau \mu_k^\delta.
$$

---

[1] If $\nabla F(x_k, \xi_k) + \mu_k(x_k - x_0) = 0$, then we can draw a new sample of ξ_k to satisfy the relation.

Combining this relation with (38), we obtain (37). Next, we show that the maximum eigenvalue of B_k is bounded. Let $Trace(\cdot)$ denote the trace of a matrix. Taking trace from both sides of (36) and summing up over index j, we obtain for $i := \lceil k/2 \rceil - (m-j)$,

$$Trace(B_{k,m}) = Trace(B_{k,0}) - \sum_{j=1}^{m} Trace\left(\frac{B_{k,j-1}s_i s_i^T B_{k,j-1}}{s_i^T B_{k,j-1}s_i}\right) + \sum_{j=1}^{m} Trace\left(\frac{y_i y_i^T}{y_i^T s_i}\right)$$

$$= Trace\left(\frac{\|y_{\lceil k/2 \rceil}\|^2}{s_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil}}\mathbf{I}\right) - \sum_{j=1}^{m} \frac{\|B_{k,j-1}s_i\|^2}{s_i^T B_{k,j-1}s_i} + \sum_{j=1}^{m} \frac{\|y_i\|^2}{y_i^T s_i}$$

(39)

$$\leq n\frac{\|y_{\lceil k/2 \rceil}\|^2}{s_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil}} + \sum_{j=1}^{m} \left(L + \tau\mu_k^\delta\right) \leq (m+n)\left(L + \tau\mu_k^\delta\right),$$

where the third relation is obtained by positive-definiteness of B_k (this can be seen by induction on j, and using (36) and $B_{k,0} \succ 0$). Since $B_k = B_{k,m}$, the maximum eigenvalue of the matrix B_k is bounded by $(m+n)\left(L + \tau\mu_k^\delta\right)$. As a result,

(40)
$$\lambda_{k,\min} \geq \frac{1}{(m+n)\left(L + \tau\mu_k^\delta\right)}.$$

In the next part of the proof, we establish the bound for $\lambda_{k,\max}$. The following relation can be shown (e.g., see Lemma 3 in [20])

$$det(B_{k,m}) = det(B_{k,0}) \prod_{j=1}^{m} \frac{s_i^T y_i}{s_i^T B_{k,j-1}s_i}, \quad \text{for } i := \lceil k/2 \rceil - (m-j).$$

Multiplying and dividing by $s_i^T s_i$, using the strong convexity of the function h, and invoking (37) and the result of (39), we obtain

$$det(B_k) = det\left(\frac{y_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil}}{s_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil}}\mathbf{I}\right)\prod_{j=1}^{m}\left(\frac{s_i^T y_i}{s_i^T s_i}\right)\left(\frac{s_i^T s_i}{s_i^T B_{k,j-1}s_i}\right)$$

$$\geq \left(\frac{y_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil}}{s_{\lceil k/2 \rceil}^T y_{\lceil k/2 \rceil}}\right)^n \prod_{j=1}^{m} \tau\mu_k^\delta \left(\frac{s_i^T s_i}{s_i^T B_{k,j-1}s_i}\right)$$

(41)
$$\geq \left(\tau\mu_k^\delta\right)^{(n+m)}\prod_{j=1}^{m}\frac{1}{(m+n)\left(L + \tau\mu_k^\delta\right)} = \frac{\left(\tau\mu_k^\delta\right)^{(n+m)}}{(m+n)^m\left(L + \tau\mu_k^\delta\right)^m}.$$

Let $\alpha_{k,1} \leq \alpha_{k,2} \leq \ldots \leq \alpha_{k,n}$ be the eigenvalues of B_k sorted non-decreasingly. Note that since $B_k \succ 0$, all the eigenvalues are positive. Taking (39) and (41) into account, and employing Lemma 6, we obtain

$$\alpha_{1,k} \geq \frac{(n-1)!\left(\tau\mu_k^\delta\right)^{(n+m)}}{(m+n)^{n+m-1}\left(L + \tau\mu_k^\delta\right)^{n+m-1}}.$$

This relation and that $\alpha_{k,1} = \lambda_{k,\max}^{-1}$ imply that

(42)
$$\lambda_{k,\max} \leq \frac{(m+n)^{n+m-1}\left(L + \tau\mu_k^\delta\right)^{n+m-1}}{(n-1)!\left(\tau\mu_k^\delta\right)^{(n+m)}}.$$

25

Therefore, from (40) and (42) and that μ_k is non-increasing, we conclude that (20) holds for k as well. Next, we show $H_k y_{\lceil k/2 \rceil} = s_{\lceil k/2 \rceil}$. From (36), for $j = m$ we obtain

$$B_{k,m} = B_{k,m-1} - \frac{B_{k,m-1} s_{\lceil k/2 \rceil} s_{\lceil k/2 \rceil}^T B_{k,m-1}}{s_{\lceil k/2 \rceil}^T B_{k,m-1} s_{\lceil k/2 \rceil}} + \frac{y_{\lceil k/2 \rceil} y_{\lceil k/2 \rceil}^T}{y_{\lceil k/2 \rceil}^T s_{\lceil k/2 \rceil}},$$

where we used $i = \lceil k/2 \rceil - (m - m) = \lceil k/2 \rceil$. Multiplying both sides of the preceding equation by $s_{\lceil k/2 \rceil}$, and using $B_k = B_{k,m}$, we have $B_k s_k = B_{k,m-1} s_{\lceil k/2 \rceil} - B_{k,m-1} s_{\lceil k/2 \rceil} + y_{\lceil k/2 \rceil} = y_{\lceil k/2 \rceil}$. Multiplying both sides of the preceding relation by H_k and invoking $H_k = B_k^{-1}$, we conclude that $H_k y_{\lceil k/2 \rceil} = s_{\lceil k/2 \rceil}$. Therefore, we showed that the statements of (a), (b), and (c) hold for an odd k, assuming that they hold for any odd $t < k$. In a similar fashion to this analysis, it can be seen that the statements hold for $t = 2m - 1$. Thus, by induction, we conclude that the statements hold for any odd $k \geq 2m - 1$. To complete the proof, it is enough to show that (20) holds for any even $k \geq 2m$. Let $t = k - 1$. Since t is odd, relation (20) holds. Writing (20) for $k - 1$, and taking into account that $H_k = H_{k-1}$, and $\mu_k < \mu_{k-1}$, we can conclude that (20) holds for any even $k \geq 2m - 1$ and this completes the proof. $\square$

7.3. Proof of Lemma 8.

Proof. In the following, we show that the presented class of sequences satisfy each of the conditions listed in Assumption 4. Throughout, we let α denote $-(m + n)\delta$.
(a) Replacing the sequences by their given rules, we obtain

$$\gamma_k \mu_k^{2\alpha-1} = \frac{\gamma_0}{(2^b \mu_0)^{1-2\alpha}} (k+1)^{-a} (k+\kappa)^{(1-2\alpha)b} \leq \frac{\gamma_0}{(2^b \mu_0)^{1-2\alpha}} (k+1)^{-a+(1-2\alpha)b}.$$

From the assumption that $\frac{a}{b} > 1 + 2\delta(m+n)$, we obtain $-a + (1 - 2\alpha)b < 0$. Thus, the preceding term goes to zero verifying Assumption 4(a).
(b) Let k be an even number. Thus, $\kappa = 2$. From (21) we have $\mu_k = \mu_{k+1} = \frac{\mu_0 2^b}{(k+2)^b}$. Now, let k be an odd number. Again, according to (21) can write

$$\mu_{k+1} = \frac{\mu_0 2^b}{((k+1)+2)^b} < \frac{\mu_0 2^b}{(k+1)^b} = \frac{\mu_0 2^b}{(k+\kappa)^b} = \mu_k.$$

Therefore, μ_k given by (21) satisfies (15). Also, from (21) we have $\mu_k \to 0$. Thus, Assumption 4(b) holds.
(c) The given rules (21) imply that γ_k and μ_k are both non-increasing sequences. Therefore, we have $\gamma_k \mu_k \leq \gamma_0 \mu_0$ for any $k \geq 0$. So, to show that Assumption 4(c) holds, it is enough to show that $\lambda_{\min} \gamma_0 \mu_0 \leq 1$ where $\lambda_{\min}$ is given by (19). Since we assumed that $\gamma_0 \mu_0 \leq L(m+n)$, for any $\delta \in (0, 1]$, we have $\gamma_0 \mu_0 \leq (m+n)(L + \mu_0^\delta)$, implying that $\lambda_{\min} \gamma_0 \mu_0 \leq 1$ and that Assumption 4(c) holds.
(d) From (21), we can write

$$\sum_{k=0}^{\infty} \gamma_k \mu_k = \gamma_0 \mu_0 2^b \sum_{k=0}^{\infty} (k+1)^{-a}(k+\kappa)^{-b} \geq \gamma_0 \mu_0 2^b \sum_{k=0}^{\infty} (k+2)^{-(a+b)} = \infty,$$

where the last relation is due to $a + b \leq 1$. Therefore, Assumption 4(d) holds.
(e) Using (21), it follows

$$\sum_{k=0}^{\infty} \gamma_k \mu_k^2 = \gamma_0 \mu_0^2 4^b \sum_{k=0}^{\infty} (k+1)^{-a}(k+\kappa)^{-2b} \leq \gamma_0 \mu_0^2 4^b \sum_{k=0}^{\infty} (k+1)^{-(a+2b)} < \infty,$$

where the last inequality is due to $a + 2b > 1$. Therefore, Assumption 4(e) holds.
(f) From (21), we have

$$\sum_{k=0}^{\infty} \gamma_k^2 \mu_k^{2\alpha} \leq \gamma_0^2 (\mu_0 2^b)^{2\alpha} \left(\sum_{k=0}^{1} \frac{(k+\kappa)^{-2\alpha b}}{(k+1)^{2a}} + \sum_{k=2}^{\infty} \frac{(2k)^{-2\alpha b}}{k^{2a}} \right) < \infty$$

where in the first inequality, we use $\alpha < 0$ and in the last inequality, we note that $a + \alpha b = a - \delta (m + n) b > 0.5$. Therefore, Assumption 4(f) is verified. □

7.4. Proof of Lemma 9.

Proof. Throughout, we let α denote $-\delta(m + n)$. Assumption 5(a, b, c) and (15) have been already shown in parts (a, b, c) of the proof of Lemma 8.
(d) It suffices to show there exists K_0 such that for any $k \geq K_0$ and $0 < \beta < 1$,

$$(43) \qquad \frac{\gamma_{k-1}}{\gamma_k} \frac{\mu_k^{1-2\alpha}}{\mu_{k-1}^{1-2\alpha}} - 1 \leq \beta \lambda_{\min} \gamma_k \mu_k.$$

From (21) and the definition of α, we obtain

$$\frac{\gamma_{k-1}}{\gamma_k} \frac{\mu_k^{1-2\alpha}}{\mu_{k-1}^{1-2\alpha}} - 1 \leq \frac{\gamma_{k-1}}{\gamma_k} - 1 = \left(1 + \frac{1}{k}\right)^a - 1 = 1 + \frac{a}{k} + o\left(\frac{1}{k}\right) - 1 = \mathcal{O}\left(\frac{1}{k}\right),$$

where the first inequality is implied due to $\{\mu_k\}$ is non-increasing, and in the second equation, we used the Taylor's expansion of $\left(1 + \frac{1}{k}\right)^a$. Therefore, since the right-hand side of (43) is of the order $\frac{1}{k^{a+b}}$ and that $a + b < 1$, the preceding inequality shows that such K_0 exists for which Assumption 5(d) holds for all $0 < \beta < 1$.
(e) From (21), we have

$$\frac{\mu_k^{2-2\alpha}}{\gamma_k} = \gamma_0^{-1} \left(\mu_0 2^b\right)^{2-2\alpha} (k+\kappa)^{-b(2-2\alpha)}(k+1)^a \leq \frac{\gamma_0^{-1} \left(\mu_0 2^b\right)^{2-2\alpha}}{(k+1)^{-a+(2-2\alpha)b}}$$

$$\leq \gamma_0^{-1} \left(\mu_0 2^b\right)^{2-2\alpha} = \rho,$$

where the first inequality is due to $\alpha < 0$, and the second inequality follows by the assumption $a \leq 2b(1 + \delta(m + n))$. Therefore, Assumption 5(e) is satisfied. □

REFERENCES

[1] A. Bordes, L. Bottou, and P. Gallinari, *SGD-QN: Careful quasi-Newton stochastic gradient descent*, Journal of Machine Learning Research **10** (2009), 1737–1754.

[2] L. Bottou, *Large-scale machine learning with stochastic gradient descent*, Proceedings of the 19th International Conference on Computational Statistics (Paris, France) (Y. Lechevallier and G. Saporta, eds.), Springer, 2010, pp. 177–187.

[3] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, *A stochastic Quasi-Newton method for large-scale optimization*, SIAM Journal on Optimization **26** (2016), no. 2, 1008–1031.

[4] C. Charitha, J. Dutta, and L. D. Russell, *Lagrange multipliers, (exact) regularization and error bounds for monotone variational inequalities*, Mathematical Programming **161** (2017), no. 1-2, 519–549.

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

[6] R. Fletcher, *A new approach to variable metric algorithms*, The Computer Journal **13** (1970), no. 3, 317–322.

[7] M. P. Friedlander and P. Tseng, *Exact regularization of convex programs*, SIAM Journal on Optimization **18** (2008), no. 4, 1326–1350.

[8] D. Goldfarb, *A family of variable-metric methods derived by variational means*, Mathematics of Computation **24** (1970), no. 109, 23–26.

[9] M. Gürbüzbalaban, A. Ozdaglar, and P. A. Parrilo, *On the convergence rate of incremental aggregated gradient algorithms*, SIAM Journal on Optimization **27** (2017), no. 2, 1035–1048.

[10] J. Koshal, A. Nedić, and U. V Shanbhag, *Regularized iterative stochastic approximation methods for stochastic variational inequality problems*, IEEE Transactions on Automatic Control **58** (2013), no. 3, 594–609.

[11] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, *RCV1: A new benchmark collection for text categorization research*, Journal of machine learning research **5** (2004), 361–397.

[12] D. Li and M. Fukushima, *A modified BFGS method and its global convergence in nonconvex minimization*, Journal of Computational and Applied Mathematics **129** (2001), 15–35.

[13] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Mathematical programming **45** (1989), 503–528.

[14] A. Lucchi, B. McWilliams, and T. Hofmann, *A variance reduced stochastic Newton method*, arXiv preprint arXiv:1503.08316 (2015).

[15] O. L. Mangasarian M. C. Ferris, *Finite perturbation of convex programs*, Applied Mathematics and Optimization **23** (1991), no. 1, 263–273.

[16] N. Majlesinasab, F. Yousefian, and A. Pourhabib, *Self-tuned mirror descent schemes for smooth and nonsmooth high-dimensional stochastic optimization*, IEEE Transactions on Automatic Control (2019), (to appear), doi: 10.1109/TAC.2019.2897889.

[17] O. L. Mangasarian and R. R. Meyerf, *Nonlinear perturbation of linear programs*, SIAM Journal of Control and Optimization **17** (1979), no. 6, 745–757.

[18] A. Mokhtari, M. Eisen, and A. Ribeiro, *IQN: An incremental quasi-Newton method with local superlinear convergence rate*, arXiv preprint arXiv:1702.00709, 2017.

[19] A. Mokhtari and A. Ribeiro, *RES: regularized stochastic BFGS algorithm*, IEEE Transactions on Signal Processing **62** (2014), no. 23, 6089–6104.

[20] ———, *Global convergence of online limited memory BFGS*, Journal of Machine Learning Research **16** (2015), 3151–3181.

[21] A. Nedić and S. Lee, *On stochastic subgradient mirror-descent algorithm with weighted averaging*, SIAM Journal on Optimization **24** (2014), no. 1, 84–107.

[22] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM Journal on Optimization **19** (2009), no. 4, 1574–1609.

[23] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed., Springer, New York, 2006.

[24] B.T. Polyak, *Introduction to optimization*, Optimization Software, Inc., New York, 1987.

[25] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Statistics **22** (1951), 400–407.

[26] M. Schmidt, N. L. Roux, and F. Bach, *Minimizing finite sums with the stochastic average gradient*, Mathematical Programming **162** (2017), no. 1-2, 83–112.

[27] N. N. Schraudolph, J. Yu, and S. Gunter, *A stochastic quasi-Newton method for online convex optimization*, In Proc. 11th Intl. Conf. on Artificial Intelligence and Statistics (AIstats) (2007), 433–440.

[28] P. Tseng and S. Yun, *Block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization*, Journal of Optimization Theory and Applications **140** (2009), no. 3, 513–535.

[29] N. D. Vanli, M. Gürbüzbalaban, and A. Ozdaglar, *Global convergence rate of incremental aggregated gradient methods for nonsmooth problems*, The 55th Conference on Decision and Control (CDC), IEEE, 2016, pp. 173–178.

[30] X. Wang, S. Ma, and W. Liu, *Stochastic quasi-Newton methods for nonconvex stochastic optimization*, SIAM Journal on Optimization **27** (2017), no. 2, 927–956.

[31] F. Yousefian, A. Nedić, and U. V. Shanbhag, *On stochastic gradient and subgradient methods with adaptive steplength sequences*, Automatica **48** (2012), no. 1, 56–67, An extended version of the paper available at: http://arxiv.org/abs/1105.4549.

[32] ———, *Self-tuned stochastic approximation schemes for non-Lipschitzian stochastic multiuser optimization and nash games*, IEEE Transactions on Automatic Control **61** (2016), no. 7, 1753–1766.

[33] ———, *Stochastic quasi-Newton methods for non-strongly convex problems: Convergence and rate analysis*, IEEE 55th Conference on Decision and Control (CDC), IEEE, 2016, pp. 4496–4503.

[34] ———, *On smoothing, regularization, and averaging in stochastic approximation methods for stochastic variational inequality problems*, Mathematical Programming (Series B.) **165** (2017), no. 1, 391–431.