IMA Journal of Numerical Analysis (2018) **00**, 1–34
doi:10.1093/imanum/dry009

Exact and inexact subsampled Newton methods for optimization

Raghu Bollapragada
Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA

Richard H. Byrd
Department of Computer Science, University of Colorado, Boulder, CO, USA

and

Jorge Nocedal*
Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA
*Corresponding author: nocedal@eecs.northwestern.edu

[Received on 17 April 2017; revised on 01 November 2017]

The paper studies the solution of stochastic optimization problems in which approximations to the gradient and Hessian are obtained through subsampling. We first consider Newton-like methods that employ these approximations and discuss how to coordinate the accuracy in the gradient and Hessian to yield a superlinear rate of convergence in expectation. The second part of the paper analyzes an inexact Newton method that solves linear systems approximately using the conjugate gradient (CG) method, and that samples the Hessian and not the gradient (the gradient is assumed to be exact). We provide a complexity analysis for this method based on the properties of the CG iteration and the quality of the Hessian approximation, and compare it with a method that employs a stochastic gradient iteration instead of the CG method. We report preliminary numerical results that illustrate the performance of inexact subsampled Newton methods on machine learning applications based on logistic regression.

Keywords: machine learning; subsampling; stochastic optimization.

1. Introduction

In this paper, we study subsampled Newton methods for stochastic optimization. These methods employ approximate gradients and Hessians, through sampling, in order to achieve efficiency and scalability. Additional economy of computation is obtained by solving linear systems inexactly at every iteration, i.e., by implementing inexact Newton methods. We study the convergence properties of (exact) Newton methods that approximate both the Hessian and gradient, as well as the complexity of inexact Newton methods that subsample only the Hessian and use the conjugate gradient (CG) method to solve linear systems.

The optimization problem of interest arises in machine learning applications, but with appropriate modifications is relevant to other stochastic optimization settings including simulation optimization (Amaran *et al.,* 2014; Fu *et al.,* 2015). We state the problem as

$$\min_{w\in\mathbb{R}^d} F(w) = \int f(w; x, y)\, \mathrm{d}P(x, y), \tag{1.1}$$

where f is the composition of a prediction function (parametrized by a vector $w \in \mathbb{R}^d$) and a smooth loss function, and (x, y) are the input–output pairs with joint probability distribution $P(x, y)$. We call F the *expected risk*.

In practice, one does not have complete information about $P(x, y)$, and therefore one works with data $\{(x^i, y^i)\}$ drawn from the distribution P. One may view an optimization algorithm as being applied directly to the expected risk (1.1), or given N data points, as being applied to the *empirical risk*

$$R(w) = \frac{1}{N}\sum_{i=1}^{N} f(w; x^i, y^i).$$

To simplify the notation, we define $F_i(w) = f(w; x^i, y^i)$, and thus we write

$$R(w) = \frac{1}{N}\sum_{i=1}^{N} F_i(w) \tag{1.2}$$

in the familiar finite-sum form arising in many applications beyond machine learning (Bertsekas, 1995). In this paper, we consider both objective functions, F and R.

The iteration of a subsampled Newton method for minimizing F is given by

$$w_{k+1} = w_k + \alpha_k p_k, \tag{1.3}$$

where p_k is the solution of the *Newton equations*

$$\nabla^2 F_{S_k}(w_k) p_k = -\nabla F_{X_k}(w_k). \tag{1.4}$$

Here, the subsampled gradient and Hessian are defined as

$$\nabla F_{X_k}(w_k) = \frac{1}{|X_k|}\sum_{i\in X_k} \nabla F_i(w_k), \qquad \nabla^2 F_{S_k}(w_k) = \frac{1}{|S_k|}\sum_{i\in S_k} \nabla^2 F_i(w_k), \tag{1.5}$$

where the sets $X_k, S_k \subset \{1, 2, \ldots\}$ index sample points (x^i, y^i) drawn at random from the distribution P. We refer to X_k and S_k as the gradient and Hessian samples—even though they only refer to indices of the samples. The choice of the sequences $\{X_k\}$ and $\{S_k\}$ gives rise to distinct algorithms, and our goal is to identify the most promising instances, in theory and in practice.

In the first part of the paper, we consider Newton methods in which the linear system (1.4) is solved *exactly*, and we identify conditions on $\{S_k\}$ and $\{X_k\}$ under which linear or superlinear rates of convergence are achieved. Exact Newton methods are practical when the number of variables d is not too large, or when the structure of the problem allows a direct factorization of the Hessian $\nabla^2 F_{S_k}$ in time linear in the number of variables d.

For most large-scale applications, however, forming the Hessian $\nabla^2 F_{S_k}(w_k)$ and solving the linear system (1.4) accurately are prohibitively expensive, and one has to compute an approximate solution in $O(d)$ time using an iterative linear solver that only requires Hessian-vector products (and not the Hessian itself). Methods based on this strategy are sometimes called *inexact Hessian-free Newton methods*. In the second part of the paper, we study how to balance the accuracy of this linear solver with the sampling

technique used for the Hessian so as to obtain an efficient algorithm. In doing so, we pay particular attention to the properties of two iterative linear solvers for (1.4), namely the CG method and a stochastic gradient iteration (SGI).

It is generally accepted that in the context of deterministic optimization and when the matrix in (1.4) is the exact Hessian, the CG method is the iterative solver of choice due to its notable convergence properties. However, subsampled Newton methods provide a different setting where other iterative methods could be more effective. For example, solving (1.4) using an SGI has the potential advantage that the sample S_k in (1.4) can be changed at every iteration, in contrast with the Newton-CG method where it is essential to fix the sample S_k throughout the CG iteration.

It is then natural to ask: which of the two methods, Newton-CG or Newton-SGI, is more efficient, and how should this be measured? Following Agarwal *et al.* (2016), we phrase this question by asking how much computational effort does each method require in order to yield a given local rate of convergence—specifically, a linear rate with convergence constant of 1/2.

1.1 *Related work*

Subsampled gradient and Newton methods have recently received much attention. Friedlander & Schmidt (2012) and Byrd *et al.* (2012) analyze the rate at which X_k should increase so that the subsampled gradient method (with fixed steplength) converges linearly to the solution of strongly convex problems. Byrd *et al.* (2012) also provide work-complexity bounds for their method and report results of experiments with a subsampled Newton-CG method, whereas Friedlander & Schmidt (2012) study the numerical performance of L-BFGS using gradient sampling techniques. Martens (2010) proposes a subsampled Gauss–Newton method for training deep neural networks, and focuses on the choice of the regularization parameter. None of these papers provide a convergence analysis for subsampled Newton methods.

Pasupathy *et al.* (2015) consider sampling rates in a more general setting. Given a deterministic algorithm—that could have linear, superlinear or quadratic convergence—they analyze the stochastic analogue that subsamples the gradient, and identify optimal sampling rates for several families of algorithms. Erdogdu & Montanari (2015) study a Newton-like method, where the Hessian approximation is obtained by first subsampling the true Hessian and then computing a truncated eigenvalue decomposition. Their method employs a full gradient and is designed for problems, where d is not so large that the cost of iteration, namely $O(Nd + |S|d^2)$, is affordable.

Roosta–Khorasani & Mahoney (2016a, 2016b) derive global and local convergence rates for subsampled Newton methods with various sampling rates used for gradient and Hessian approximations. Our convergence results are similar to theirs, except that they employ matrix concentration inequalities (Tropp & Wright, 2010) and state progress at each iteration in probability—whereas we do so in expectation. The results in Roosta–Khorasani & Mahoney (2016) go beyond other studies in the literature in that they assume the objective function is strongly convex, but the individual component functions are only weakly convex, and show how to ensure that the subsampled matrices are invertible (in probability). Xu *et al.* (2016) study the effect of nonuniform sampling. They also compare the complexity of Newton-CG and Newton-SGI by estimating the amount of work required to achieve a given rate of linear convergence, as is done in this paper. However, their analysis establishes convergence rates in probability, *for one iteration*, whereas we prove convergence in expectation for the sequence of iterates.

Pilanci & Wainwright (2015) propose a Newton sketch method that approximates the Hessian via random projection matrices while employing the full gradient of the objective. The best complexity results are obtained when the projections are performed using the randomized Hadamard transform.

This method requires access to the square root of the true Hessian, which, for generalized linear models, entails access to the entire dataset at each iteration.

Agarwal *et al.* (2016) study a Newton method that aims to compute unbiased estimators of the inverse Hessian and that achieves a linear time complexity in the number of variables. Although they present the method as one that computes a power expansion of the Hessian inverse, they show that it can also be interpreted as a subsampled Newton method where the step computation is performed inexactly using an SGI.

Sampling-based Newton methods have been used in many applications such as imaging problems (Herrmann & Li, 2012; Calatroni *et al.,* 2017), and in training deep and recurrent neural networks (Martens & Sutskever, 2012).

1.2 *Notation*

We denote the variables of the optimization problem by $w \in \mathbb{R}^d$, and a minimizer of the objective F as w^*. Throughout the paper, we use $\|\cdot\|$ to represent the ℓ_2 vector norm or its induced matrix norm. The notation $A \preceq B$ means that $B - A$ is a symmetric and positive semidefinite matrix.

2. Subsampled Newton methods

The problem under consideration in this section is the minimization of expected risk (1.1). The general form of an (exact) subsampled Newton method for this problem is given by

$$w_{k+1} = w_k - \alpha_k \nabla^2 F_{S_k}^{-1}(w_k) \nabla F_{X_k}(w_k), \tag{2.1}$$

where $\nabla^2 F_{S_k}(w_k)$ and $\nabla F_{X_k}(w_k)$ are defined in (1.5). We assume that the sets $\{X_k\}$, $\{S_k\} \subset \{1, 2, \ldots\}$ are chosen independently (with or without replacement).

The steplength α_k in (2.1) is chosen to be a constant or computed by a backtracking line search; we do not consider the case of diminishing steplengths, $\alpha_k \to 0$, which leads to a sublinear rate of convergence. It is therefore clear that, in order to obtain convergence to the solution, the sample size X_k must increase, and in order to obtain a rate of convergence that is faster than linear, the Hessian sample S_k must also increase. We now investigate the rules for controlling those samples. The main set of assumptions made in this paper is as follows.

Assumption 2.1

A1 **(Bounded Eigenvalues of Hessians)** The function F is twice continuously differentiable and any subsampled Hessian is positive definite with eigenvalues lying in a positive interval (that depends on the sample size). That is, for any integer β and any set $S \subset \{1, 2, \ldots\}$ with $|S| = \beta$, there exist positive constants μ_β, L_β such that

$$\mu_\beta I \preceq \nabla^2 F_S(w) \preceq L_\beta I, \quad \forall w \in \mathbb{R}^d. \tag{2.2}$$

Moreover, there exist constants $\bar{\mu}, \bar{L}$ such that

$$0 < \bar{\mu} \leq \mu_\beta \quad \text{and} \quad L_\beta \leq \bar{L} < \infty, \quad \text{for all } \beta \in \mathbb{N}. \tag{2.3}$$

The smallest and largest eigenvalues corresponding to the objective F are denoted by μ, L (with $0 < \mu, L < \infty$), i.e.,

$$\mu I \preceq \nabla^2 F(w) \preceq LI, \quad \forall w \in \mathbb{R}^d. \tag{2.4}$$

A2 **(Bounded Variance of Sample Gradients)** The trace of the covariance matrix of the individual sample gradients is uniformly bounded, i.e., there exists a constant v such that

$$\operatorname{tr}\left(\operatorname{Cov}\left(\nabla F_i(w)\right)\right) \leq v^2, \qquad \forall w \in \mathbb{R}^d. \tag{2.5}$$

A3 **(Lipschitz Continuity of Hessian)** The Hessian of the objective function F is Lipschitz continuous, i.e., there is a constant $M > 0$ such that

$$\|\nabla^2 F(w) - \nabla^2 F(z)\| \leq M\|w - z\|, \qquad \forall w, z \in \mathbb{R}^d. \tag{2.6}$$

A4 **(Bounded Variance of Hessian Components)** There is a constant σ such that, for all component Hessians, we have

$$\|\mathbb{E}[(\nabla^2 F_i(w) - \nabla^2 F(w))^2]\| \leq \sigma^2, \qquad \forall w \in \mathbb{R}^d. \tag{2.7}$$

We let w^* denote the unique minimizer of F. In practice, most problems are regularized and therefore assumption 2.1(A1) is satisfied. Concerning A2 and A4, variances are always bounded for a finite sum problem, and it is standard to assume so for the expected risk problem. Assumption 2.1(A3) is used only to establish superlinear convergence and it is natural in that context.

2.1 *Global linear convergence*

We now show that for the Newton method (2.1) to enjoy an R-linear rate of convergence the gradient sample size must be increased at a geometric rate, i.e., $|X_k| = \eta^k$ for some $\eta > 1$. On the other hand, the subsampled Hessian need not be accurate, and thus it suffices to keep samples S_k of constant size, $|S_k| = \beta \geq 1$. The following result, in which the steplength α_k in (2.1) is constant, is a simple extension of well-known results (see, e.g., Byrd *et al.*, 2012; Friedlander & Schmidt, 2012; Pasupathy *et al.*, 2015), but we include the proof for the sake of completeness. We assume that the set X_k is drawn uniformly at random so that at every iteration $\mathbb{E}[\nabla F_{X_k}(w_k)] = \nabla F(w_k)$. We also assume that the sets X_k and S_k are chosen independently of each other.

THEOREM 2.2 Suppose that Assumptions 2.1(A1)–(A2) hold. Let $\{w_k\}$ be the iterates generated by iteration (2.1) with any w_0, where $|X_k| = \eta^k$ for some $\eta > 1$, and $|S_k| = \beta \geq 1$ is constant. Then, if the steplength satisfies $\alpha_k = \alpha = \frac{\mu_\beta}{L}$, we have that

$$\mathbb{E}[F(w_k) - F(w^*)] \leq C\hat{\rho}^k, \tag{2.8}$$

where

$$C = \max\left\{F(w_0) - F(w^*), \frac{v^2 L_\beta}{\mu\mu_\beta}\right\} \quad \text{and} \quad \hat{\rho} = \max\left\{1 - \frac{\mu\mu_\beta}{2LL_\beta}, \frac{1}{\eta}\right\}. \tag{2.9}$$

Proof. Let $\mathbb{E}_k$ denote the conditional expectation at iteration k for all possible sets X_k. Then for any given S_k,

$$\begin{aligned}
\mathbb{E}_k\left[F(w_{k+1})\right] &\leq F(w_k) - \alpha \nabla F(w_k)^T \nabla^2 F_{S_k}^{-1}(w_k) \mathbb{E}_k[\nabla F_{X_k}(w_k)] + \frac{L\alpha^2}{2} \mathbb{E}_k[\|\nabla^2 F_{S_k}^{-1}(w_k) \nabla F_{X_k}(w_k)\|^2] \\
&= F(w_k) - \alpha \nabla F(w_k)^T \nabla^2 F_{S_k}^{-1}(w_k) \nabla F(w_k) + \frac{L\alpha^2}{2} \|\mathbb{E}_k[\nabla^2 F_{S_k}^{-1}(w_k) \nabla F_{X_k}(w_k)]\|^2 \\
&\quad + \frac{L\alpha^2}{2} \mathbb{E}_k[\|\nabla^2 F_{S_k}^{-1}(w_k) \nabla F_{X_k}(w_k) - \mathbb{E}_k[\nabla^2 F_{S_k}^{-1}(w_k) \nabla F_{X_k}(w_k)]\|^2] \\
&= F(w_k) - \alpha \nabla F(w_k)^T \left(\nabla^2 F_{S_k}^{-1}(w_k) - \frac{L\alpha}{2} \nabla^2 F_{S_k}^{-2}(w_k) \right) \nabla F(w_k) \\
&\quad + \frac{L\alpha^2}{2} \mathbb{E}_k[\|\nabla^2 F_{S_k}^{-1}(w_k) \nabla F_{X_k}(w_k) - \nabla^2 F_{S_k}^{-1}(w_k) \nabla F(w_k)\|^2] \\
&\leq F(w_k) - \alpha \nabla F(w_k)^T \nabla^2 F_{S_k}^{-1/2}(w_k) \left(I - \frac{L\alpha}{2} \nabla^2 F_{S_k}^{-1}(w_k) \right) \nabla^2 F_{S_k}^{-1/2}(w_k) \nabla F(w_k) \\
&\quad + \frac{L\alpha^2}{2\mu_\beta^2} \mathbb{E}_k[\|\nabla F_{X_k}(w_k) - \nabla F(w_k)\|^2].
\end{aligned}$$

Now, $\{\nabla^2 F_{S_k}^{-1}\}$ is a sequence of random variables, but we can bound its eigenvalues from above and below. Therefore, we can use these eigenvalue bounds as follows:

$$\begin{aligned}
\mathbb{E}_k\left[F(w_{k+1})\right] &\leq F(w_k) - \alpha \left(1 - \frac{L\alpha}{2\mu_\beta}\right) (1/L_\beta) \|\nabla F(w_k)\|^2 + \frac{L\alpha^2}{2\mu_\beta^2} \mathbb{E}_k[\|\nabla F_{X_k}(w_k) - \nabla F(w_k)\|^2] \\
&\leq F(w_k) - \frac{\mu_\beta}{2LL_\beta} \|\nabla F(w_k)\|^2 + \frac{1}{2L} \mathbb{E}_k[\|\nabla F_{X_k}(w_k) - \nabla F(w_k)\|^2] \\
&\leq F(w_k) - \frac{\mu\mu_\beta}{LL_\beta} (F(w_k) - F(w^*)) + \frac{1}{2L} \mathbb{E}_k[\|\nabla F(w_k) - \nabla F_{X_k}(w_k)\|^2],
\end{aligned}$$

where the last inequality follows from the fact that, for any μ −strongly convex function, $\|\nabla F(w_k)\|^2 \geq 2\mu(F(w_k) - F(w^*))$. Therefore, we get

$$\mathbb{E}_k\left[F(w_{k+1}) - F(w^*)\right] \leq \left(1 - \frac{\mu\mu_\beta}{LL_\beta}\right) (F(w_k) - F(w^*)) + \frac{1}{2L} \mathbb{E}_k[\|\nabla F(w_k) - \nabla F_{X_k}(w_k)\|^2]. \quad (2.10)$$

Now, by Assumption 2.1(A2) we have that

$$\begin{aligned}\mathbb{E}_k[\|\nabla F(w_k) - \nabla F_{X_k}(w_k)\|^2] &= \mathbb{E}_k\left[\operatorname{tr}\left(\left(\nabla F(w_k) - \nabla F_{X_k}(w_k)\right)\left(\nabla F(w_k) - \nabla F_{X_k}(w_k)\right)^T\right)\right] \\ &= \operatorname{tr}\left(\operatorname{Cov}\left(\nabla F_{X_k}(w_k)\right)\right) \\ &= \operatorname{tr}\left(\operatorname{Cov}\left(\frac{1}{|X_k|}\sum_{i\in X_k}\nabla F_i(w_k)\right)\right) \\ &\leq \frac{1}{|X_k|}\operatorname{tr}(\operatorname{Cov}(\nabla F_i(w_k))) \\ &\leq \frac{v^2}{|X_k|}. \end{aligned} \tag{2.11}$$

Substituting this inequality in (2.10), we obtain

$$\mathbb{E}_k\left[F(w_{k+1}) - F(w^*)\right] \leq \left(1 - \frac{\mu\mu_\beta}{LL_\beta}\right)(F(w_k) - F(w^*)) + \frac{v^2}{2L|X_k|}. \tag{2.12}$$

We use induction for the rest of the proof, and to this end we recall the definitions of C and $\hat{\rho}$. Since $\mathbb{E}[F(w_0) - F(w^*)] \leq C$, inequality (2.8) holds for $k = 0$. Now, suppose that (2.8) holds for some k. Combining (2.12), the condition $|X_k| = \eta^k$, and the definition of $\hat{\rho}$, we have

$$\begin{aligned}\mathbb{E}[F(w_{k+1}) - F(w^*)] &\leq \left(1 - \frac{\mu\mu_\beta}{LL_\beta}\right)C\hat{\rho}^k + \frac{v^2}{2L|X_k|} \\ &= C\hat{\rho}^k\left(1 - \frac{\mu\mu_\beta}{LL_\beta} + \frac{v^2}{2LC(\hat{\rho}\eta)^k}\right) \\ &\leq C\hat{\rho}^k\left(1 - \frac{\mu\mu_\beta}{LL_\beta} + \frac{v^2}{2LC}\right) \\ &\leq C\hat{\rho}^k\left(1 - \frac{\mu\mu_\beta}{LL_\beta} + \frac{\mu\mu_\beta}{2LL_\beta}\right) \\ &= C\hat{\rho}^k\left(1 - \frac{\mu\mu_\beta}{2LL_\beta}\right) \\ &\leq C\hat{\rho}^{k+1}. \end{aligned}$$

□

If one is willing to increase the Hessian sample size as the iterations progress, then one can achieve a faster rate of convergence, as discussed next.

2.2 *Local superlinear convergence*

We now discuss how to design the subsampled Newton method (using a unit stepsize) so as to obtain superlinear convergence in a neighborhood of the solution w^*. This question is most interesting when the objective function is given by the expectation (1.1) and the indices i are chosen from an infinite set according to a probability distribution P. We will show that the sample size used for gradient estimation should increase at a rate that is *faster than geometric*, i.e., $|X_k| \geq \eta_k^k$, where $\{\eta_k\} > 1$ is an increasing sequence, whereas sample size used for Hessian estimation should increase at any rate such that $|S_k| \geq |S_{k-1}|$ and $\lim_{k\to\infty} |S_k| = \infty$.

We begin with the following result that identifies three quantities that drive the iteration. Here $\mathbb{E}_k$ denotes the conditional expectation at iteration k for all possible sets X_k and S_k.

LEMMA 2.3 Let $\{w_k\}$ be the iterates generated by algorithm (2.1) with $\alpha_k = 1$, and suppose that Assumptions 2.1(A1)–(A3) hold. Then for each k,

$$\mathbb{E}_k[\|w_{k+1} - w^*\|] \leq \frac{1}{\mu_{|S_k|}} \left[\frac{M}{2} \|w_k - w^*\|^2 + \mathbb{E}_k \left[\left\| \left(\nabla^2 F_{S_k}(w_k) - \nabla^2 F(w_k) \right) (w_k - w^*) \right\| \right] + \frac{v}{\sqrt{|X_k|}} \right]. \tag{2.13}$$

Proof. We have that the expected distance to the solution after the kth step is given by

$$\begin{aligned} \mathbb{E}_k[\|w_{k+1} - w^*\|] &= \mathbb{E}_k[\|w_k - w^* - \nabla^2 F_{S_k}^{-1}(w_k) \nabla F_{X_k}(w_k)\|] \\ &= \mathbb{E}_k \left[\left\| \nabla^2 F_{S_k}^{-1}(w_k) \left(\nabla^2 F_{S_k}(w_k)(w_k - w^*) - \nabla F(w_k) - \nabla F_{X_k}(w_k) + \nabla F(w_k) \right) \right\| \right] \\ &\leq \frac{1}{\mu_{|S_k|}} \mathbb{E}_k \left[\left\| \left(\nabla^2 F_{S_k}(w_k) - \nabla^2 F(w_k) \right) (w_k - w^*) + \nabla^2 F(w_k)(w_k - w^*) - \nabla F(w_k) \right\| \right] \\ &\quad + \frac{1}{\mu_{|S_k|}} \mathbb{E}_k \left[\left\| \nabla F_{X_k}(w_k) - \nabla F(w_k) \right\| \right]. \end{aligned} \tag{2.14}$$

Therefore,

$$\begin{aligned} \mathbb{E}_k[\|w_{k+1} - w^*\|] \leq & \underbrace{\frac{1}{\mu_{|S_k|}} \|\nabla^2 F(w_k)(w_k - w^*) - \nabla F(w_k)\|}_{\text{Term 1}} \\ & + \underbrace{\frac{1}{\mu_{|S_k|}} \mathbb{E}_k \left[\left\| \left(\nabla^2 F_{S_k}(w_k) - \nabla^2 F(w_k) \right) (w_k - w^*) \right\| \right]}_{\text{Term 2}} \\ & + \underbrace{\frac{1}{\mu_{|S_k|}} \mathbb{E}_k[\|\nabla F_{X_k}(w_k) - \nabla F(w_k)\|]}_{\text{Term 3}}. \end{aligned} \tag{2.15}$$

For Term 1, we have by Lipschitz continuity of the Hessian (2.6),

$$\begin{aligned}\frac{1}{\mu_{|S_k|}}\|\nabla^2 F(w_k)(w_k - w^*) - \nabla F(w_k)\| &\leq \frac{1}{\mu_{|S_k|}}\|w_k - w^*\| \left\| \int_{t=0}^{1} [\nabla^2 F(w_k) - \nabla^2 F(w_k + t(w^* - w_k))]\, \mathrm{d}t \right\| \\ &\leq \frac{1}{\mu_{|S_k|}}\|w_k - w^*\| \int_{t=0}^{1} \|\nabla^2 F(w_k) - \nabla^2 F(w_k + t(w^* - w_k))\|\, \mathrm{d}t \\ &\leq \frac{1}{\mu_{|S_k|}}\|w_k - w^*\|^2 \int_{t=0}^{1} Mt\, \mathrm{d}t \\ &= \frac{M}{2\mu_{|S_k|}}\|w_k - w^*\|^2.\end{aligned}$$

Term 3 represents the error in the gradient approximation. By Jensen's inequality we have that,

$$\left(\mathbb{E}_k[\|\nabla F(w_k) - \nabla F_{X_k}(w_k)\|]\right)^2 \leq \mathbb{E}_k[\|\nabla F(w_k) - \nabla F_{X_k}(w_k)\|^2].$$

We have shown in (2.11) that $\mathbb{E}_k[\|\nabla F_{X_k}(w_k) - \nabla F(w_k)\|^2] \leq v^2/|X_k|$, which concludes the proof. □

Let us now consider Term 2 in (2.13), which represents the error due to Hessian subsampling. In order to prove convergence, we need to bound this term as a function of the Hessian sample size $|S_k|$. The following lemma shows that this error is inversely related to the square root of the sample size.

LEMMA 2.4 Suppose that Assumptions 2.1(A1) and (A4) hold. Then

$$\mathbb{E}_k\left[\left\|\left(\nabla^2 F_{S_k}(w_k) - \nabla^2 F(w_k)\right)(w_k - w^*)\right\|\right] \leq \frac{\sigma}{\sqrt{|S_k|}}\|w_k - w^*\|, \tag{2.16}$$

where σ is defined in Assumption 2.1(A4).

Proof. Let us define $Z_S = \nabla^2 F_S(w)(w - w^*)$ and $Z = \nabla^2 F(w)(w - w^*)$, so that

$$\left\|\left(\nabla^2 F_S(w) - \nabla^2 F(w)\right)(w - w^*)\right\| = \|Z_S - Z\|.$$

(For convenience we drop the iteration index k in this proof.) We also write $Z_S = \frac{1}{|S|}\sum Z_i$, where $Z_i = \nabla^2 F_i(w)(w - w^*)$, and note that each Z_i is independent. By Jensen's inequality we have,

$$\begin{aligned}(\mathbb{E}[\|Z_S - Z\|])^2 &\leq \mathbb{E}[\|Z_S - Z\|^2] \\ &= \mathbb{E}\left[\mathrm{tr}\left((Z_S - Z)(Z_S - Z)^T\right)\right] \\ &= \mathrm{tr}(\mathrm{Cov}(Z_S)) \\ &= \mathrm{tr}\left(\mathrm{Cov}\left(\frac{1}{|S|}\sum_{i\in S} Z_i\right)\right) \\ &\leq \frac{1}{|S|}\mathrm{tr}(\mathrm{Cov}(Z_i)) \\ &= \frac{1}{|S|}\mathrm{tr}\left(\mathrm{Cov}(\nabla^2 F_i(w)(w - w^*))\right).\end{aligned}$$

Now, we have by (2.7) and (2.4)

$$\operatorname{tr}\left(\operatorname{Cov}(\nabla^2 F_i(w)(w-w^*))\right) = (w-w^*)^T \mathbb{E}[(\nabla^2 F_i(w) - \nabla^2 F(w))^2](w-w^*)$$
$$\leq \sigma^2 \|w-w^*\|^2.$$

□

We note that in the literature on subsampled Newton methods (Erdogdu & Montanari, 2015; Roosta–Khorasani & Mahoney, 2016b) it is common to use matrix concentration inequalities to measure the accuracy of Hessian approximations. In Lemma 2.4, we measure instead the error along the vector $w_k - w^*$, which gives a more precise estimate.

Combining Lemma 2.3 and Lemma 2.4, and recalling (2.3), we obtain the following linear-quadratic bound

$$\mathbb{E}_k[\|w_{k+1} - w^*\|] \leq \frac{M}{2\bar{\mu}}\|w_k - w^*\|^2 + \frac{\sigma\|w_k - w^*\|}{\bar{\mu}\sqrt{|S_k|}} + \frac{v}{\bar{\mu}\sqrt{|X_k|}}. \tag{2.17}$$

It is clear that in order to achieve a superlinear convergence rate, it is not sufficient to increase the sample size $|X_k|$ at a geometric rate, because that would decrease the last term in (2.17) at a linear rate; thus $|X_k|$ must be increased at a rate that is faster than geometric. From the middle term we see that sample size $|S_k|$ should also increase, and can do so at any rate, provided $|S_k| \to \infty$. To bound the first term, we introduce the following assumption on the second moments of the distance of iterates to the optimum.

Assumption 2.5

B1 **(Bounded Moments of Iterates)** There is a constant $\gamma > 0$ such that for any iterate w_k generated by algorithm (2.1) we have

$$\mathbb{E}[||w_k - w^*||^2] \leq \gamma(\mathbb{E}[||w_k - w^*||])^2. \tag{2.18}$$

This assumption seems, at a first glance, to be very restrictive. But we note that it is imposed on non-negative numbers, and that it is less restrictive than assuming that the iterates are bounded (for all possible choices of the random variables); see e.g., Babanezhad *et al.* (2015) and the references therein. For a general stochastic optimization problem, this assumption that the iterates are bounded implies that B1 holds.

We now show local superlinear convergence when a unit steplength is employed. Superlinear convergence can only be established if the steplength is 1 (or converges to 1). The analysis here applies both to the case where steplength of 1 is used from a sufficiently close starting point, and to the case where some test such as sufficient decrease is used to decide when to start using unit steplength after using a strategy such as the one described above.

Theorem 2.6 (**Superlinear convergence**) Let $\{w_k\}$ be the iterates generated by algorithm 2.1 with stepsize $\alpha_k = \alpha = 1$. Suppose that Assumptions 2.1(A1)–(A4) and B1 hold and that for all k:

(i) $|X_k| \geq |X_0|\,\eta_k^k$, with $|X_0| \geq \left(\frac{6v\gamma M}{\bar{\mu}^2}\right)^2$, $\eta_k > \eta_{k-1}$, $\eta_k \to \infty$ and $\eta_1 > 1$.

(ii) $|S_k| > |S_{k-1}|$, with $\lim_{k\to\infty} |S_k| = \infty$, and $|S_0| \geq \left(\frac{4\sigma}{\bar{\mu}}\right)^2$.

Then, if the starting point satisfies

$$\|w_0 - w^*\| \leq \frac{\bar{\mu}}{3\gamma M},$$

we have that $\mathbb{E}[\|w_k - w^*\|] \to 0$ at an R-superlinear rate, i.e., there is a positive sequence $\{\tau_k\}$ such that

$$\mathbb{E}[\|w_k - w^*\|] \leq \tau_k \quad \text{and} \quad \tau_{k+1}/\tau_k \to 0.$$

Proof. We establish the result by showing that, for all k,

$$\mathbb{E}[\|w_k - w^*\|] \leq \frac{\bar{\mu}}{3\gamma M}\tau_k, \tag{2.19}$$

where

$$\tau_{k+1} = \max\left\{\tau_k\rho_k, \eta_{k+1}^{-(k+1)/4}\right\}, \quad \tau_0 = 1, \quad \rho_k = \frac{\tau_k}{6} + \frac{1}{4}\sqrt{\frac{|S_0|}{|S_k|}} + \frac{1}{2\eta_k^{k/4}}. \tag{2.20}$$

We use induction to show (2.19). Note that the base case, $k = 0$, is trivially satisfied. Let us assume that the result is true for iteration k, so that

$$\frac{3\gamma M}{\bar{\mu}}\mathbb{E}[\|w_k - w_*\|] \leq \tau_k.$$

Let us now consider iteration $k + 1$. Using (2.17), the bounds for the sample sizes given in conditions (i)–(ii), (2.20) and (2.18), we get

$$\begin{aligned}
\mathbb{E}\left[\mathbb{E}_k\left[\|w_{k+1} - w^*\|\right]\right] &\leq \mathbb{E}\left[\frac{M}{2\bar{\mu}}\|w_k - w^*\|^2 + \frac{\sigma\|w_k - w^*\|}{\bar{\mu}\sqrt{|S_k|}} + \frac{v}{\bar{\mu}\sqrt{|X_k|}}\right] \\
&\leq \frac{\gamma M}{2\bar{\mu}}(\mathbb{E}[\|w_k - w^*\|])^2 + \frac{\sigma\mathbb{E}[\|w_k - w^*\|]}{\bar{\mu}\sqrt{|S_k|}} + \frac{v}{\bar{\mu}\sqrt{|X_k|}} \\
&\leq \frac{\bar{\mu}}{3\gamma M}\tau_k\left(\frac{\tau_k}{6}\right) + \frac{\bar{\mu}}{3\gamma M}\tau_k\left(\frac{1}{4}\sqrt{\frac{|S_0|}{|S_k|}}\right) + \frac{\bar{\mu}}{3\gamma M}\left(\frac{1}{2\sqrt{\eta_k^k}}\right) \\
&= \frac{\bar{\mu}}{3\gamma M}\tau_k\left[\frac{\tau_k}{6} + \frac{1}{4}\sqrt{\frac{|S_0|}{|S_k|}} + \frac{1}{2\tau_k\sqrt{\eta_k^k}}\right] \\
&\leq \frac{\bar{\mu}}{3\gamma M}\tau_k\left[\frac{\tau_k}{6} + \frac{1}{4}\sqrt{\frac{|S_0|}{|S_k|}} + \frac{1}{2\eta_k^{k/4}}\right] \\
&= \frac{\bar{\mu}}{3\gamma M}\tau_k\rho_k \;\leq\; \frac{\bar{\mu}}{3\gamma M}\tau_{k+1},
\end{aligned}$$

which proves (2.19).

To prove R-superlinear convergence, we show that the sequence τ_k converges superlinearly to 0. First, we use induction to show that $\tau_k < 1$. For the base case $k = 1$ we have,

$$\rho_0 = \frac{\tau_0}{6} + \frac{1}{4} + \frac{1}{2} = \frac{1}{6} + \frac{1}{4} + \frac{1}{2} = \frac{11}{12} < 1,$$

$$\tau_1 = \max\left\{\tau_0\rho_0, \eta_1^{(-1/4)}\right\} = \max\left\{\rho_0, \eta_1^{-1/4}\right\} < 1.$$

Now, let us assume that $\tau_k < 1$ for some $k > 1$. By the fact that $\{|S_k|\}$ and $\{\eta_k\}$ are increasing sequences, we obtain

$$\rho_k = \frac{\tau_k}{6} + \frac{1}{4}\sqrt{\frac{|S_0|}{|S_k|}} + \frac{1}{2\eta_k^{k/4}} \leq \frac{1}{6} + \frac{1}{4} + \frac{1}{2} = \frac{11}{12} < 1, \tag{2.21}$$

$$\tau_{k+1} = \max\left\{\tau_k\rho_k, \eta_{k+1}^{-(k+1)/4}\right\} \leq \max\left\{\rho_k, \eta_1^{-(k+1)/4}\right\} < 1,$$

which proves that $\tau_k < 1$ for all $k > 1$.

Moreover, since $\rho_k \leq 11/12$, we see from the first definition in (2.20) (and the fact that $\eta_k \to \infty$) that the sequence $\{\tau_k\}$ converges to zero. This implies by the second definition in (2.20) that $\{\rho_k\} \to 0$.

Using these observations, we conclude that

$$\begin{aligned}
\lim_{k\to\infty} \frac{\tau_{k+1}}{\tau_k} &= \lim_{k\to\infty} \frac{\max\left\{\tau_k\rho_k, \eta_{k+1}^{-(k+1)/4}\right\}}{\tau_k} \\
&= \lim_{k\to\infty} \max\left\{\rho_k, \frac{\eta_{k+1}^{-(k+1)/4}}{\tau_k}\right\} \\
&\leq \lim_{k\to\infty} \max\left\{\rho_k, \left(\frac{\eta_k}{\eta_{k+1}}\right)^{k/4} \frac{1}{\eta_{k+1}^{1/4}}\right\} \\
&\leq \lim_{k\to\infty} \max\left\{\rho_k, \frac{1}{\eta_{k+1}^{1/4}}\right\} = 0.
\end{aligned}$$

□

The constants 6 and 4 in assumptions (i)–(ii) of this theorem were chosen for the sake of simplicity, to avoid introducing general parameters, and other values could be chosen.

Let us compare this result with those established in the literature. We established superlinear convergence in expectation. In contrast the results in the study by Roosta–Khorasani & Mahoney (2016b) show a rate of decrease in the error at a given iteration with certain probability $1 - \delta$. Concatenating such statements does not give convergence guarantees of the overall sequence with high probability. We could ask whether the approach described in the studies by Erdogdu & Montanari (2015), Roosta–Khorasani & Mahoney (2016b) and Xu *et al.* (2016), together with Assumption 2.5(B1), could be used to prove convergence in expectation. This is not the case because one cannot guarantee that the Hessian approximation is invertible, and hence one would still need additional assumptions

to prove convergence in expectation. In contrast, our approach can be used to prove convergence in probability because of the assumptions that variances are bounded allows one to use the Chebyshev inequality to derive a probability bound, and then derive a convergence result in probability.

Pasupathy *et al.* (2015) use a different approach to show that the entire sequence of iterates converges in expectation. They assume that the 'deterministic analog' has a *global superlinear rate* of convergence, a rather strong assumption. In conclusion, all the superlinear results just mentioned are useful and shed light into the properties of subsampled Newton methods, but none of them seem to be definitive.

3. Inexact Newton-CG method

We now consider inexact subsampled Newton methods in which the Newton equations (1.4) are solved approximately. A natural question that arises in this context is the relationship between the accuracy in the solution of (1.4), the size of the sample S_k and the rate of convergence of the method. Additional insights are obtained by analyzing the properties of specific iterative solvers for the system (1.4), and we focus here on the CG method. We provide a complexity analysis for an inexact subsampled Newton-CG method, and in Section 4, compare it with competing approaches.

In this section, we assume that the Hessian is subsampled, but the gradient is not. Since computing the full (exact) gradient is more realistic when the objective function is given by the finite sum (1.2), we assume throughout this section that the objective is R. The iteration therefore has the form

$$w_{k+1} = w_k + p_k^r \tag{3.1}$$

where p_k^r is the approximate solution obtained after applying r steps of the CG method to the $d \times d$ linear system

$$\nabla^2 R_{S_k}(w_k)p = -\nabla R(w_k), \quad \text{with} \quad \nabla^2 R_{S_k}(w_k) = \frac{1}{|S_k|}\sum_{i\in S_k} \nabla^2 F_i(w_k). \tag{3.2}$$

We assume that the number r of CG steps performed at every iteration is constant, as this facilitates our complexity analysis which is phrased in terms of $|S_k|$ and r. (Later on, we consider an alternative setting where the accuracy in the linear system solve is controlled by a residual test.) Methods in which the Hessian is subsampled, but the gradient is not are sometimes called *semistochastic*, and several variants have been studied in the studies by Agarwal *et al.* (2016), Byrd *et al.* (2011), Pilanci & Wainwright (2015) and Roosta–Khorasani & Mahoney (2016b).

A sharp analysis of the Newton-CG method (3.1)–(3.2) is difficult to perform because the convergence rate of the CG method varies at every iteration depending on the spectrum $\{\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_d\}$ of the positive definite matrix $\nabla^2 R_{S_k}(w_k)$. For example, after computing r steps of the CG method applied to the linear system in (3.2), the iterate p^r satisfies

$$||p^r - p^*||_A^2 \leq \left(\frac{\lambda_{d-r} - \lambda_1}{\lambda_{d-r} + \lambda_1}\right)^2 ||p^0 - p^*||_A^2, \quad \text{with} \quad A = \nabla^2 R_{S_k}(w_k). \tag{3.3}$$

Here p^* denotes the exact solution and $\|x\|_A^2 \stackrel{\text{def}}{=} x^T Ax$. In addition, one can show that CG will terminate in t iterations, where t denotes the number of distinct eigenvalues of $\nabla^2 R_{S_k}(w_k)$, and also show that the method does not approach the solution at a steady rate.

Since an analysis based on the precise bound (3.3) is complex, we make use of the *worst case behavior of CG* (Golub & Van Loan, 1989) which is given by

$$\|p^r - p^*\|_A \leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^r \|p^0 - p^*\|_A, \tag{3.4}$$

where $\kappa(A)$ denotes the condition number of A. Using this bound, we can establish the following linear-quadratic bound. Here the sample S_k is allowed to vary at each iteration but its size, $|S_k|$, is assumed constant.

LEMMA 3.1 Let $\{w_k\}$ be the iterates generated by the inexact Newton method (3.1)–(3.2), where $|S_k| = \beta$ and the direction p_k^r is the result of performing $r < d$ CG iterations on the system (3.2). Suppose Assumptions 2.1(A1), (A3) and (A4) hold. Then,

$$\mathbb{E}_k[\|w_{k+1} - w^*\|] \leq C_1\|w_k - w^*\|^2 + \left(\frac{C_2}{\sqrt{\beta}} + C_3\theta^r\right)\|w_k - w^*\|, \tag{3.5}$$

where

$$C_1 = \frac{M}{2\mu_\beta}, \quad C_2 = \frac{\sigma}{\mu_\beta}, \quad C_3 = \frac{2L}{\mu_\beta}\sqrt{\delta(\beta)}, \quad \theta = \left(\frac{\sqrt{\delta(\beta)}-1}{\sqrt{\delta(\beta)}+1}\right), \quad \delta(\beta) = \frac{L_\beta}{\mu_\beta}. \tag{3.6}$$

Proof. We have that

$$\begin{aligned}\mathbb{E}_k[\|w_{k+1} - w^*\|] &= \mathbb{E}_k[\|w_k - w^* + p_k^r\|] \\ &\leq \underbrace{\mathbb{E}_k[\|w_k - w^* - \nabla^2 R_{S_k}^{-1}(w_k)\nabla R(w_k)\|]}_{\text{Term 4}} + \underbrace{\mathbb{E}_k[\|p_k^r + \nabla^2 R_{S_k}^{-1}(w_k)\nabla R(w_k)\|]}_{\text{Term 5}}.\end{aligned} \tag{3.7}$$

Term 4 was analyzed in the previous section where the objective function is F, i.e., where the iteration is defined by (2.1) so that (2.14) holds. In our setting, we have that Term 3 in (2.15) is zero (since the gradient is not sampled) and hence, from (2.13),

$$\mathbb{E}_k[\|w_k - w^* - \nabla^2 R_{S_k}^{-1}(w_k)\nabla R(w_k)\|] \leq \frac{M}{2\mu_\beta}\|w_k - w^*\|^2 + \frac{1}{\mu_\beta}\mathbb{E}_k\left[\left\|\left(\nabla^2 R_{S_k}(w_k) - \nabla^2 R(w_k)\right)(w_k - w^*)\right\|\right]. \tag{3.8}$$

Recalling Lemma 2.4 (with R replacing F) and the definitions (3.6), we have

$$\mathbb{E}_k[\|w_k - w^* - \nabla^2 R_{S_k}^{-1}(w_k)\nabla R(w_k)\|] \leq C_1\|w_k - w^*\|^2 + \frac{C_2}{\sqrt{\beta}}\|w_k - w^*\|. \tag{3.9}$$

Now, we analyze Term 5, which is the residual in the CG solution after r iterations. Assuming for simplicity that the initial CG iterate is $p_k^0 = 0$, we obtain from (3.4)

$$\|p_k^r + \nabla^2 R_{S_k}^{-1}(w_k)\nabla R(w_k)\|_A \leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^r \|\nabla^2 R_{S_k}^{-1}(w_k)\nabla R(w_k)\|_A,$$

where $A = \nabla^2 R_{S_k}(w_k)$. To express this in terms of unweighted norms, note that if $\|a\|_A^2 \leq \|b\|_A^2$, then

$$\lambda_1\|a\|^2 \leq a^T A a \leq b^T A b \leq \lambda_d\|b\|^2 \implies \|a\| \leq \sqrt{\kappa(A)}\|b\|.$$

Therefore, from Assumption 2.1(A1)

$$\begin{aligned}
\mathbb{E}_k[\|p_k^r + \nabla^2 R_{S_k}^{-1}(w_k)\nabla R(w_k)\|] &\leq 2\sqrt{\kappa(A)}\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^r \mathbb{E}_k[\|\nabla^2 R_{S_k}^{-1}(w_k)\nabla R(w_k)\|] \\
&\leq 2\sqrt{\kappa(A)}\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^r \|\nabla R(w_k)\|\,\mathbb{E}_k[\|\nabla^2 R_{S_k}^{-1}(w_k)\|] \\
&\leq \frac{2L}{\mu_\beta}\sqrt{\kappa(A)}\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^r \|w_k - w^*\| \\
&= C_3\theta^r\|w_k - w^*\|, \qquad (3.10)
\end{aligned}$$

where the last step follows from the fact that, by the definition of A, we have $\mu_\beta \leq \lambda_1 \leq \cdots \leq \lambda_d \leq L_\beta$, and hence $\kappa(A) \leq L_\beta/\mu_\beta = \delta(\beta)$. □

We now use Lemma 3.1 to determine the number of Hessian samples $|S|$ and the number of CG iterations r that guarantee a *given* rate of convergence. Specifically, we require a linear rate with constant 1/2, in a neighborhood of the solution. This will allow us to compare our results with those in the study by Agarwal *et al.* (2016). We recall that C_1 is defined in (3.6) and γ in (2.18).

THEOREM 3.2 Suppose that Assumptions 2.1(A1), (A3), (A4) and 2.5(B1) hold. Let $\{w_k\}$ be the iterates generated by inexact Newton-CG method (3.1)–(3.2), with

$$|S_k| = \beta \geq \frac{64\sigma^2}{\bar{\mu}^2}, \qquad (3.11)$$

and suppose that the number of CG steps performed at every iteration satisfies

$$r \geq \log\left(\frac{16L}{\mu_\beta}\sqrt{\delta(\beta)}\right)\frac{1}{\log\left(\frac{\sqrt{\delta(\beta)}+1}{\sqrt{\delta(\beta)}-1}\right)}.$$

Then, if $||w_0 - w^*|| \leq \min\{\frac{1}{4C_1}, \frac{1}{4\gamma C_1}\}$, we have

$$\mathbb{E}[||w_{k+1} - w^*||] \leq \tfrac{1}{2}\mathbb{E}[||w_k - w^*||]. \qquad (3.12)$$

Proof. By the definition of C_2 given in (3.6) and (3.11), we have that $C_2/\sqrt{\beta} \leq 1/8$. Now,

$$\begin{aligned} C_3\theta^r &= \frac{2L}{\mu_\beta}\sqrt{\delta(\beta)}\left(\frac{\sqrt{\delta(\beta)}-1}{\sqrt{\delta(\beta)}+1}\right)^r \\ &\leq \frac{2L}{\mu_\beta}\sqrt{\delta(\beta)}\left(\frac{\sqrt{\delta(\beta)}-1}{\sqrt{\delta(\beta)}+1}\right)^{\left(\frac{\log\left(\frac{16L}{\mu_\beta}\sqrt{\delta(\beta)}\right)}{\log\left(\frac{\sqrt{\delta(\beta)}+1}{\sqrt{\delta(\beta)}-1}\right)}\right)} \\ &= \frac{2L}{\mu_\beta}\sqrt{\delta(\beta)}\frac{1}{\frac{16L}{\mu_\beta}\sqrt{\delta(\beta)}} \quad = \frac{1}{8}. \end{aligned}$$

We use induction to prove (3.12). For the base case we have from Lemma 3.1,

$$\begin{aligned} \mathbb{E}[\|w_1 - w^*\|] &\leq C_1\|w_0 - w^*\|\|w_0 - w^*\| + \left(\frac{C_2}{\sqrt{\beta}} + C_3\theta^r\right)\|w_0 - w^*\| \\ &\leq \tfrac{1}{4}\|w_0 - w^*\| + \tfrac{1}{4}\|w_0 - w^*\| \\ &= \tfrac{1}{2}\|w_0 - w^*\|. \end{aligned}$$

Now suppose that (3.12) is true for k^{th} iteration. Then,

$$\begin{aligned} \mathbb{E}\left[\mathbb{E}_k\left[\|w_{k+1} - w^*\|\right]\right] &\leq C_1\mathbb{E}[\|w_k - w^*\|^2] + \left(\frac{C_2}{\sqrt{\beta}} + C_3\theta^r\right)\mathbb{E}[\|w_k - w^*\|] \\ &\leq \gamma C_1\mathbb{E}[\|w_k - w^*\|]\,\mathbb{E}[\|w_k - w^*\|] + \left(\frac{C_2}{\sqrt{\beta}} + C_3\theta^r\right)\mathbb{E}[\|w_k - w^*\|] \\ &\leq \tfrac{1}{4}\mathbb{E}[\|w_k - w^*\|] + \tfrac{1}{4}\mathbb{E}[\|w_k - w^*\|] \\ &= \tfrac{1}{2}\mathbb{E}[\|w_k - w^*\|]. \end{aligned}$$

□

We note that condition (3.11) may require a value of β greater than N. In this case the proof of Theorem 3.2 is clearly still valid if we sample with replacement, but this is a wasteful strategy since it achieves the bound $|S_k| > N$ by repeating samples. If we wish to sample without replacement in this case, we can set $\beta = N$. Then our Hessian approximation is exact and the C_2 term is zero, so the proof still goes through and Theorem 3.2 holds.

This result was established using the worst case complexity bound of the CG method. We know, however, that CG converges to the solution in at most d steps. Hence, the bound on the maximum number of iterations needed to obtain a linear rate of convergence with constant 1/2 is

$$r = \min\left\{d, \frac{\log\left(16L\sqrt{\delta(\beta)}/\mu_\beta\right)}{\log\left(\frac{\sqrt{\delta(\beta)}+1}{\sqrt{\delta(\beta)}-1}\right)}\right\}. \tag{3.13}$$

This bound is still rather pessimistic in practice since most problems do not give rise to the worst case behavior of the method.

Convergence rate controlled by residual test. In many practical implementations of inexact Newton methods, the CG iteration is stopped based on the norm of the residual in the solution of the linear system (1.4), rather than on a prescribed maximum number r of CG iterations (Dembo *et al.,* 1982). For the system (3.2), this residual-based termination test is given by

$$\|\nabla^2 R_{S_k}(w_k)p_k^r + \nabla R(w_k)\| \leq \zeta \|\nabla R(w_k)\|, \tag{3.14}$$

where $\zeta < 1$ is a control parameter. Lemma 3.1 still applies in this setting, but with a different constant $C_3\theta^r$. Specifically, Term 5 in (3.7) is modified as follows:

$$\begin{aligned}\mathbb{E}_k[\|p_k^r + \nabla^2 R_{S_k}^{-1}(w_k)\nabla R(w_k)\|] &\leq \mathbb{E}_k[\|\nabla^2 R_{S_k}^{-1}(w_k)\|\|\nabla^2 R_{S_k}(w_k)p_k^r - \nabla R(w_k)\|] \\ &\leq \frac{\zeta}{\mu_\beta}\|\nabla R(w_k)\| \\ &\leq \frac{L\zeta}{\mu_\beta}\|w_k - w^*\|.\end{aligned}$$

Hence, comparing with (3.10), we now have that $C_3\theta^r = \frac{L}{\mu_\beta}\zeta$. To obtain linear convergence with constant 1/2, we must impose a bound on the parameter ζ, so as to match the analysis in Theorem 3.2, where we required that $C_3\theta^r \leq \frac{1}{8}$. This condition is satisfied if

$$\zeta \leq \frac{\mu_\beta}{8L}.$$

Thus, the parameter ζ must be inversely proportional to a quantity related to the condition number of the Hessian.

We conclude this section by remarking that the results presented in this section may not reflect the full power of the subsampled Newton-CG method since we assumed the worst case behavior of CG, and as noted in (3.3), the per-iteration progress can be much faster than in the worst case.

4. Comparison with other methods

We now ask whether the CG method is, in fact, an efficient linear solver when employed in the inexact subsampled Newton-CG method (3.1)–(3.2), or whether some other iterative linear solver could be preferable. Specifically, we compare CG with a semi-stochastic gradient iteration that is described below; we denote the variant of (3.1)–(3.2) that uses the SGI iteration to solve linear systems as the Newton-SGI method. Following Agarwal *et al.* (2016), we measure efficiency by estimating the total number of Hessian-vector products required to achieve a local linear rate of convergence with convergence constant 1/2 (the other costs of the algorithms are identical).

To present the complexity results of this section we introduce the following definitions of condition numbers:

$$\hat{\kappa} = \frac{\bar{L}}{\mu}, \quad \hat{\kappa}^{\max} = \frac{\bar{L}}{\bar{\mu}} \quad \text{and } \kappa = \frac{L}{\mu}.$$

Newton-CG method. Since each CG iteration requires 1 Hessian-vector product, every iteration of the inexact subsampled Newton-CG method requires $\beta \times r$ Hessian-vector products, where $\beta = |S_k|$ and r are the number of CG iterations performed.

By the definitions in Assumption 2.1, we have that σ^2 is bounded by a multiple of $\bar{L}^2$. Therefore, recalling the definition (3.6) we have that the sample size stipulated in Theorem 3.2 and (3.6) satisfies

$$\beta = O(C_2^2) = O(\sigma^2/\bar{\mu}^2) = O((\hat{\kappa}^{\max})^2).$$

Now, from (3.13) the number of CG iterations satisfies the bound

$$\begin{aligned} r &= O\left(\min\left\{d, \frac{\log((L/\mu_\beta)\sqrt{\delta(\beta)})}{\log\left(\frac{\sqrt{\delta(\beta)}+1}{\sqrt{\delta(\beta)}-1}\right)}\right\}\right) \\ &= O\left(\min\left\{d, \sqrt{\hat{\kappa}^{\max}}\log(\hat{\kappa}^{\max})\right\}\right), \end{aligned}$$

where the last equality is by the fact that $\delta(\beta) \leq \hat{\kappa}^{\max}$ and $L \leq \bar{L}$. Therefore, the number of Hessian-vector products required by the Newton-CG method to yield a linear convergence rate with constant of 1/2 is

$$O(\beta r) = O\left((\hat{\kappa}^{\max})^2 \min\left\{d, \sqrt{\hat{\kappa}^{\max}}\log(\hat{\kappa}^{\max})\right\}\right). \tag{4.1}$$

Newton-SGI method. To motivate this method, we first note that a step of the classical Newton method is given by the minimizer of the quadratic model

$$Q(p) = R(w_k) + \nabla R(w_k)^T p + \frac{1}{2} p^T \nabla^2 R(w_k) p. \tag{4.2}$$

We could instead minimize Q using the gradient method,

$$p_{t+1} = p_t - \nabla Q(p_t) = (I - \nabla^2 R(w_k))p_t - \nabla R(w_k),$$

but the cost of the Hessian-vector product in this iteration is expensive. Therefore, one can consider the semi-stochastic gradient iteration

$$p_{t+1} = (I - \nabla^2 R_i(w_k))p_t - \nabla R(w_k), \tag{4.3}$$

where the index i is chosen at random from $\{1, \ldots, N\}$. We define the Newton-SGI method by $w_{k+1} = w_k + p_r$, where p_r is the iterate obtained after applying r iterations of (4.3).

Agarwal *et al.* (2016) analyze a method they call LiSSA that is related to this Newton-SGI method. Although they present their method as one based on a power expansion of the inverse Hessian, they note in (Agarwal *et al.*, 2016, Section 4.2) that, if the outer loop in their method is disabled (by setting $S_1 = 1$), then their method is equivalent to our Newton-SGI method. They provide a complexity bound for the more general version of the method in which they compute S_2 iterations of (4.3), repeat this S_1 times, and then calculate the average of all the solutions to define the new iterate. They provide a bound, in

probability, for one step of their overall method, whereas our bounds for Newton-CG are in expectation. In spite of these differences, it is interesting to compare the complexity of the two methods.

The number of Hessian-vector products for LiSSA (which is given $O(S_1 S_2)$ in their notation) is

$$O((\hat{\kappa}^{\max})^2 \hat{\kappa} \log(\hat{\kappa}) \log(d)). \tag{4.4}$$

When comparing this estimate with (4.1) we observe that the Newton-CG bounds depend on the square root of a condition number, whereas Newton-SGI depends on the condition number itself. Furthermore, Newton-CG also has an improvement of $\log(d)$ because our proof techniques avoid the use of matrix concentration bounds.

We note in passing that certain implicit assumptions are made about the algorithms discussed above when the objective is given by the finite sum R. In subsampled Newton methods, it is assumed that the number of subsamples is less than the number of examples n. This implies that for all these methods one makes the implicit assumption that $n > \kappa^2$. We should also note that in all the stochastic second order methods, the number of samples required by the theory is κ^2, but in practice a small number of samples suffice to give good performance. This suggests that the theory could be improved and that techniques other than concentration bounds might help in achieving this.

Work complexity to obtain an ϵ-accurate solution. Table 1 compares a variety of methods in terms of the total number of gradient and Hessian-vector products required to obtain an ϵ-accurate solution. The results need to be interpreted with caution as the convergence rate of the underlying methods differs in nature, as we explain below. Therefore, Table 1 should be regarded mainly as summary of results in the literature and not as a simple way to rank methods. In stating these results, we assume that the cost of a Hessian-vector product is same as the cost of a gradient evaluation, which is realistic in many (but not all) applications.

TABLE 1 *Time complexity to obtain an ϵ-accurate solution. Comparison of the Newton-CG (Inexact) method analyzed in this paper with other well-known methods. The third column reports orders of magnitude*

Method	Convergence	Time to reach ϵ-accurate solution	Reference
SG	Global	$\frac{d\omega\kappa^2}{\epsilon}$	Bottou & Le Cun (2005)
DSS	Global	$\frac{d\nu\kappa}{\mu\epsilon}$	Byrd *et al.* (2012)
GD	Global	$nd\kappa \log(\frac{1}{\epsilon})$	Nocedal & Wright (1999)
Newton	Local	$nd^2 \log\log(\frac{1}{\epsilon})$	Nocedal & Wright (1999)
Newton-CG (Exact)	Local	$(n + (\hat{\kappa}^{\max})^2 d) d \log(\frac{1}{\epsilon})$	[This paper]
Newton-CG (Inexact)	Local	$(n + (\hat{\kappa}^{\max})^2 \sqrt{\hat{\kappa}^{\max}}) d \log(\frac{1}{\epsilon})$	[This paper]
LiSSA	Local	$(n + (\hat{\kappa}^{\max})^2 \hat{\kappa}) d \log(\frac{1}{\epsilon})$	Agarwal *et al.* (2016)

In Table 1, SG is the classical stochastic gradient method with diminishing step sizes. The complexity results of SG do not depend on n but depend on κ^2, and are inversely proportional to ϵ due to its sub-linear rate of convergence. The constant ω is the trace of the inverse Hessian times a covariance matrix; see Bottou *et al.* (2008). DSS is subsampled gradient method, where the Hessian is the identity (i.e., no Hessian subsampling is performed) and the gradient sample size $|X_k|$ increases at

a geometric rate. The complexity bounds for this method are also independent of n, and depend on κ rather than κ^2 as in SG.

GD and Newton are the classical deterministic gradient descent and Newton methods. Newton-CG (Exact and Inexact) are the subsampled Newton methods discussed in this paper. In these methods, $(\hat{\kappa}^{\max})^2$ samples are used for Hessian sampling, and the number of inner CG iterations is of order $O(d)$ for the exact method, and $O(\sqrt{\hat{\kappa}^{\max}})$ for the inexact method. LiSSA is the method proposed in Agarwal *et al.* (2016), wherein the inner solver is a semi-stochastic gradient iteration; i.e., it is similar to our Newton-SGI method, but we quote the complexity results from Agarwal *et al.* (2016). The bounds for LiSSA differ from those of Newton-CG in a square root of a condition number.

A note of caution: Table 1 lists methods with different types of convergence results. For GD and Newton, convergence is deterministic; for SG, DSS and Newton-CG (Exact & Inexact), convergence is in expectation; and for LiSSA the error (for a given iteration) is in probability. The definition of an ϵ-accurate solution also varies. For all the first order methods (SG, DSS, GD) it represents accuracy in function values; for all the second-order methods (Newton, Newton-CG, LiSSA) it represents accuracy in the iterates ($\|w - w^*\|$). Although for a strongly convex function, these two measures are related, they involve a different constant in the complexity estimates.

5. Numerical experiments

We conducted numerical experiments to illustrate the performance of the inexact subsampled Newton methods discussed in Section 3. We consider binary classification problems, where the training objective function is given by the logistic loss with ℓ_2 regularization:

$$R(w) = \frac{1}{N}\sum_{i=1}^{N} \log(1 + \exp(-y^i w^T x^i)) + \frac{\lambda}{2}\|w\|^2. \tag{5.1}$$

The regularization parameter is chosen as $\lambda = \frac{1}{N}$. The iterative linear solvers, CG and SGI, require Hessian-vector products, which are easily computed.

Table 2 summarizes the datasets used for the experiments. Some of these datasets divide the data into training and testing sets; for the rest, we randomly divide the data so that the training set constitutes 70% of the total. In Table 2, N denotes the total number of examples in a dataset, including training and testing points.

The following methods were tested in our experiments.

GD. The gradient descent method $w_{k+1} = w_k - \alpha_k \nabla R(w_k)$.

Newton. The exact Newton method $w_{k+1} = w_k + \alpha_k p_k$, where p_k is the solution of the system $\nabla^2 R(w_k)p_k = -\nabla R(w_k)$ computed to high accuracy by the CG method.

Newton-CG. The inexact subsampled Newton-CG method $w_{k+1} = w_k + \alpha_k p_k^r$, where p_k^r is an inexact solution of the linear system

$$\nabla^2 R_{S_k}(w_k)p_k = -\nabla R(w_k) \tag{5.2}$$

TABLE 2 *A description of binary datasets used in the experiments*

Data set	Data points N	Variables d	Reference
MNIST	70000	784	LeCun *et al.* (2010)
Covertype	581012	54	Blackard & Dean (1999)
Mushroom	8124	112	Lichman (2013)
Synthetic	10000	50	Mukherjee *et al.* (2013)
CINA	16033	132	CINA (2008)
Gisette	7000	5000	Guyon *et al.* (2004)

computed using the CG method. The set S_k varies at each iteration, but its cardinality $|S_k|$ is constant.

Newton-SGI. The inexact subsampled Newton-SGI method $w_{k+1} = w_k + \alpha_k p_k$, where p_k is an inexact solution of (5.2) computed by the SGI (4.3).

All these methods implement an Armijo back tracking line search to determine the steplength α_k, employ the full gradient $\nabla R(w)$ and differ in their use of second-order information. In the Newton-CG method, the CG iteration is terminated when one of the following two conditions is satisfied:

$$\|\nabla^2 R_{S_k}(w_k)p_k^j + \nabla R(w_k)\| \leq \zeta \|\nabla R(w_k)\| \qquad \text{or} \quad j = \max_{cg}, \tag{5.3}$$

where j indices the CG iterations. The parameters in these tests were set as $\zeta = 0.01$ and $\max_{cg} = 10$, which are common values in practice. These parameter values were chosen beforehand and were not tuned to our test set.

In all the figures below, *training error* is defined as $R(w) - R(w^*)$, where R is defined in terms of the data points given by the training set; *testing error* is defined as $R(w)$, without the regularization term (and using the data points from the test set).

We begin by reporting results on the `Synthetic` dataset, as they are representative of what we have observed in our experiments. Results on the other datasets are given in the appendix. In Fig. 1, we compare GD, Newton and three variants of the Newton-CG method with sample sizes $|S_k|$ given as 5%, 10% and 50% of the training data. We generate two plots: (a) Training error vs. iterations, and (b) Training error vs. *number of effective gradient evaluations*, by which we mean that each Hessian-vector product is equated with a gradient and function evaluation. Figure 2 we plot testing error vs. time. Note that the dominant computations in these experiments are gradient evaluations, Hessian-vector products and function evaluations in the line search.

Results comparing GD, Newton and Newton-CG on the rest of the test problems are given in the appendix.

In the second set of experiments, reported in Figs 3 and 4, we compare Newton-CG and Newton-SGI, again on the `Synthetic` dataset. We note that Newton-SGI is similar to the method denoted as LiSSA in Agarwal *et al.* (2016). That method contains an outer iteration that averages iterates, but in the tests reported in Agarwal *et al.* (2016), the outer loop was disabled (by setting their parameter S_1 to 1), giving rise to the Newton-SGI iteration. To guarantee convergence of the SGI iteration (4.3) (which uses a unit steplength) one must ensure that the spectral norm of the Hessian for each data point is strictly less than 1; we enforced this by rescaling the data. To determine the number of inner iterations

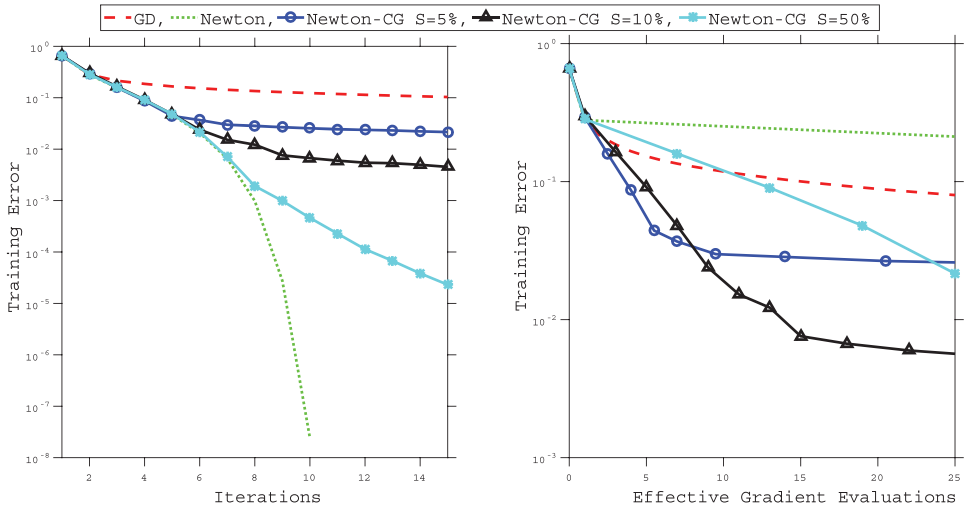

FIG. 1. **Synthetic Dataset**: Performance of the inexact subsampled Newton method (Newton-CG), using three values of the sample size, and of the GD and Newton methods. Left: Training Error vs. Iterations; Right: Training Error vs. Effective Gradient Evaluations.

in SGI, we proceeded as follows. First, we chose *one* sample size $\beta = |S|$ for the Newton-CG method, as well as the maximum number $\max_{cg}$ of CG iterations. Then, we set the number of SGI iterations to be $It = \beta \times \max_{cg}$, so that the per iteration number of Hessian-vector products in the two methods is similar. We observe from Fig. 3 that Newton-CG and Newton-SGI perform similarly in terms of effective gradient evaluations, but note from Fig. 4 the Newton-SGI has higher computing times due to

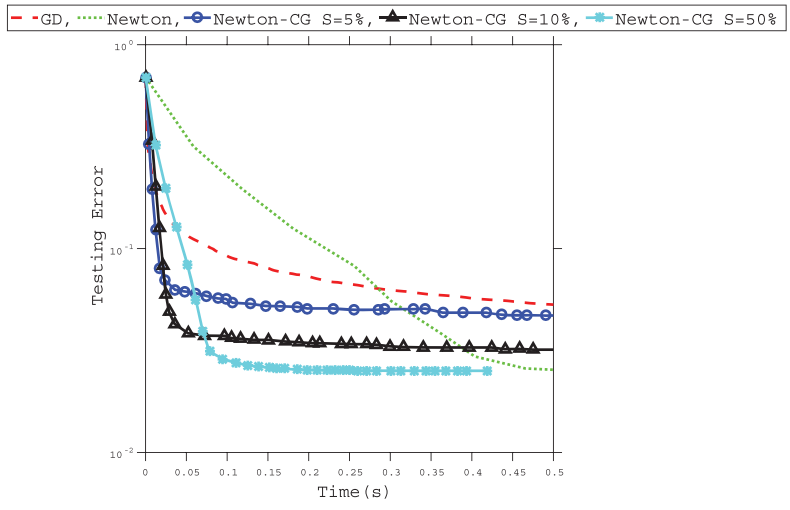

FIG. 2. **Synthetic Dataset**: Comparison of the five methods in Fig. 1, this time plotting Testing Error vs. CPU Time.

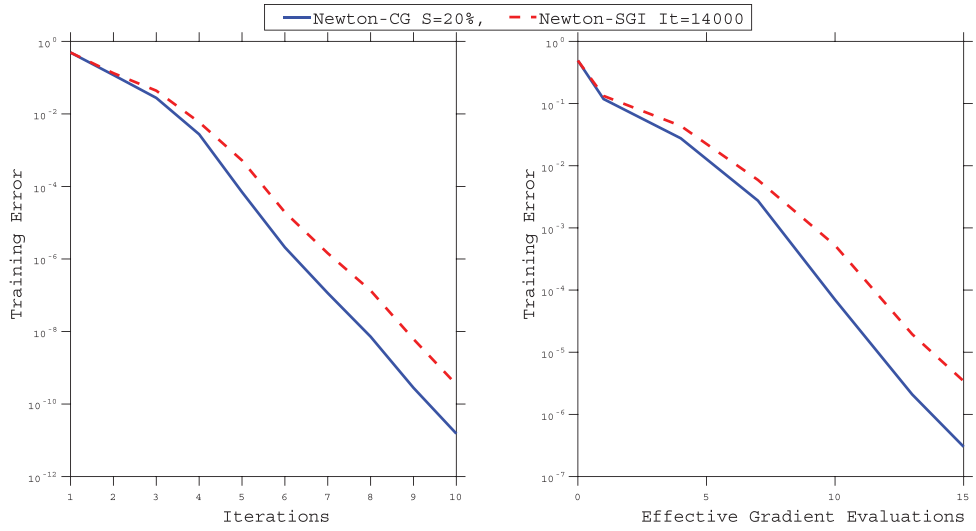

FIG. 3. **Synthetic Dataset (scaled)**: Comparison of Newton-CG and Newton-SGI. Left: Training Error vs. Iterations; Right: Training Error vs. Effective Gradient Evaluations. Here `It` denotes the number of iterations of the SGI algorithm (4.3) performed at every iteration of Newton-SGI.

the additional communication cost involved in individual Hessian-vector products. Similar results can be observed for the test problems in the appendix.

In the third set of experiments, reported in Figs 5 and 6, we compare the Newton-CG and Newton-SGI methods on the datasets *without scaling*, i.e., the spectral norm of the Hessians is now allowed to be

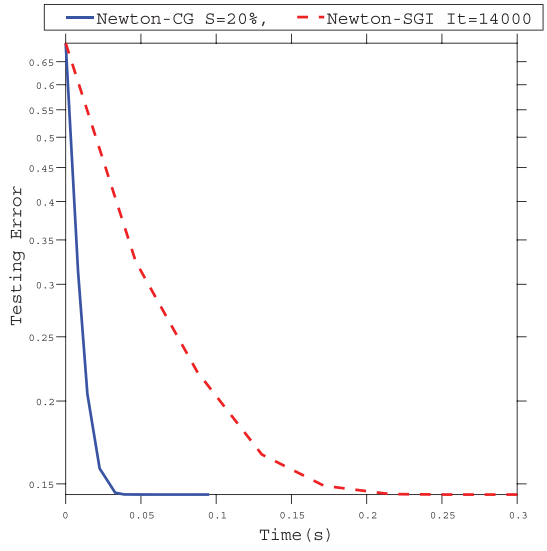

FIG. 4. **Synthetic Dataset (scaled)**: Comparison of Newton-CG with Newton-SGI, this time plotting Testing Error vs. Time.

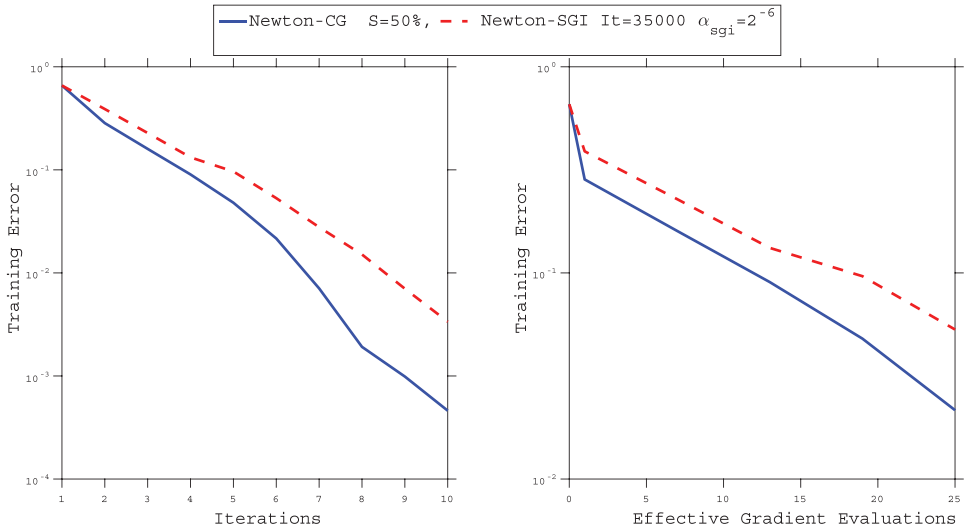

FIG. 5. **Synthetic Dataset (unscaled)**: Comparison of Newton-CG with Newton-SGI. Left: Training Error vs. Iterations; Right: Training Error vs. Effective Gradient Evaluations. The parameter α_{sgi} refers to the steplength in (5.4).

greater than 1. To ensure convergence, we modify the SGI iteration (4.3) by incorporating a step-length parameter α_{sgi}, yielding the following iteration:

$$p_{t+1} = p_t - \alpha_{sgi}\nabla Q_i(p_t) = (I - \alpha_{sgi}\nabla^2 F_i(w_k))p_t - \alpha_{sgi}\nabla R(w_k). \tag{5.4}$$

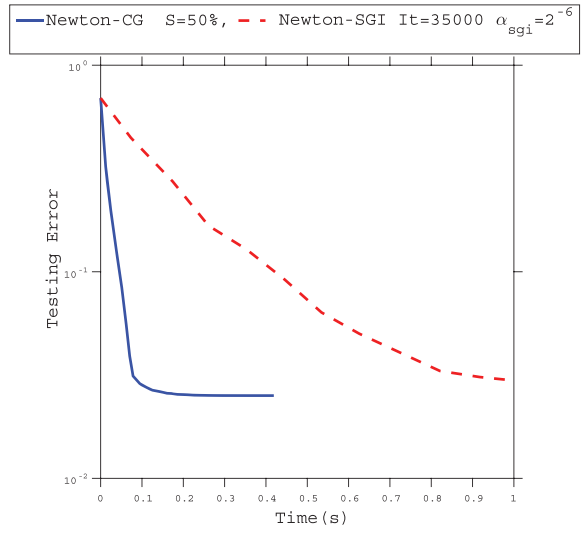

FIG. 6. **Synthetic Dataset (unscaled)**: Comparison of Newton-CG with Newton-SGI, this time plotting Testing Error vs. Time.

The steplength parameter α_{sgi} was chosen as the value in $\{2^{-20}, \ldots, 2^3\}$ that gives best overall performance.

Results comparing Newton-CG and Newton-SGI on the rest of the test datasets are given in the appendix. Overall, the numerical experiments reported in this paper suggest that the inexact subsampled Newton methods are quite effective in practice, and that there does not seem to be a concrete benefit of using the SGI iteration over the CG method.

6. Final remarks

Subsampled Newton methods (Martens, 2010; Byrd *et al.,* 2011; Byrd & Chin, 2012; Erdogdu & Montanari, 2015; Agarwal *et al.,* 2016; Roosta–Khorasani & Mahoney, 2016a, 2016b; Xu *et al.,* 2016) are attractive in large-scale applications due to their ability to incorporate *some* second-order information at low cost. They are more stable than first-order methods and can yield a faster rate of convergence. In this paper, we established conditions under which a method that subsamples the gradient and the Hessian enjoys a superlinear rate of convergence in expectation. To achieve this, the sample size used to estimate the gradient is increased at a rate that is faster than geometric, while the sample size for the Hessian approximation can increase at any rate.

The paper also studies the convergence properties of an inexact subsampled Newton method in which the step computation is performed by means of the CG method. As in Agarwal *et al.* (2016), Erdogdu & Montanari (2015), Pilanci & Wainwright (2015), Roosta–Khorasani & Mahoney (2016a, 2016b) and Xu *et al.* (2016) this method employs the full gradient and approximates the Hessian by subsampling. We give bounds on the total amount of work needed to achieve a given linear rate of convergence, and compare these bounds with those given in Agarwal *et al.* (2016) for an inexact Newton method that solves linear systems using an SGI. Computational work is measured by the number of evaluations of individual gradients and Hessian vector products.

Recent results on subsampled Newton methods (Erdogdu & Montanari, 2015; Roosta–Khorasani & Mahoney, 2016b; Xu *et al.* 2016) establish a rate of decrease at every iteration, in probability. The results of this paper are stronger in that we establish convergence in expectation, but we note that in order to do so we introduced assumption (2.18). Recent work on subsampled Newton methods focuses on the effect of nonuniform subsampling Xu *et al.* (2016), but in this paper we consider only uniform sampling.

The numerical results presented in this paper, although preliminary, make a good case for the value of subsampled Newton methods, and suggest that a more detailed and comprehensive investigation is worthwhile. We leave that study as a subject for future research.

Funding

Raghu Bollapragada was supported by the Office of Naval Research award N000141410313. Richard Byrd was supported by the National Science Foundation grant DMS-1620070. Jorge Nocedal was supported by the Department of Energy grant DE-FG02-87ER25047 and the National Science Foundation grant DMS-1620022.

References

Agarwal, N., Bullins, B. & Hazan, E. (2016) Second order stochastic optimization in linear time. arXiv preprint arXiv:1602.03943.

Amaran, S., Sahinidis, N. V., Sharda, B. & Bury, S. J. (2014) Simulation optimization: a review of algorithms and applications. *4OR*, **12**, 301–333.

Babanezhad, R., Ahmed, M. O., Virani, A., Schmidt, M., Konečnyà, J. & Sallinen, S. (2015) *Stop wasting my gradients: Practical svrg*. Advances in Neural Information Processing Systems (NIPS), vol. 28. Red Hook, New York: Curran Associates, Inc.

Bertsekas, D. P. (1995) *Nonlinear Programming*. Belmont, Massachusetts: Athena Scientific.

Blackard, J. A. & Dean, D. J. (1999) Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Comput. Electronics Agriculture.*, **24**, 131–151.

Bottou, L. & Le Cun, Y. (2005). On-line learning for very large datasets. *Appl. Stochastic Models Bus. Industry*, **21**, 137–151.

Bottou, L. & Bousquet, O. (2008) The tradeoffs of large scale learning. *Advances in Neural Information Processing Systems* (J. C. Platt, D. Koller, Y. Singer & S. Roweis eds) vol. 20. Cambridge, MA: MIT Press, pp. 161–168.

Byrd, R. H., Chin, G. M., Neveitt, W. & Nocedal, J. (2011) On the use of stochastic Hessian information in optimization methods for machine learning. *SIAM J. Optimization*, **21**, 977–995.

Byrd, R. H., Chin, G. M., Nocedal, J. & Wu, Y. (2012) Sample size selection in optimization methods for machine learning. *Math. Programming*, **134**, 127–155.

Calatroni, L., Chung, C., De Los Reyes, J. C., Chung, C., Schönlieb, C.-B. & Valkonen, T. (2017) Bilevel approaches for learning of variational imaging models. *Variational Methods in Imaging and Geometric Control*. Berlin, Germany: Walter de Gruyter GmbH & Co.KG, pp. 252–290.

Causality workbench team. A marketing dataset. Available at http://www.causality.inf.ethz.ch/data/CINA.html, 09 2008.

Dembo, R. S., Eisenstat, S. C. & Steihaug, T. (1982) Inexact-Newton methods. *SIAM J. Numer. Anal.*, **19**, 400–408.

Erdogdu, M. A. & Montanari, A. (2015) *Convergence rates of sub-sampled newton methods*. Advances in Neural Information Processing Systems, vol. 28. Red Hook, New York: Curran Associates, Inc. pp. 3034–3042.

Friedlander, M. P. & Schmidt, M. (2012) Hybrid deterministic-stochastic methods for data fitting. *SIAM J. Sci. Comput.*, **34**, A1380–A1405.

Fu M. *et al*. (2015) *Handbook of Simulation Optimization*, vol. 216. New York, New York: Springer.

Golub, G. H. & Van Loan, C. F. (1989) *Matrix Computations*, 2nd edn. Baltimore: Johns Hopkins University Press.

Guyon, I., Gunn, S., Ben Hur, A. & Dror, G. (2004) *Result analysis of the NIPS 2003 feature selection challenge*. Advances in Neural Information Processing Systems, vol. 17. Cambridge, MA: MIT Press, pp. 545–552.

Herrmann, F. J. & Li, X. (2012) Efficient least-squares imaging with sparsity promotion and compressive sensing. *Geophysical Prospecting*, **60**, 696–712.

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

Lichman, M. (2013) UCI machine learning repository. Available at http://archive.ics.uci.edu/ml.

Martens, J. (2010) Deep learning via Hessian-free optimization. *In Proceedings of the 27th International Conference on Machine Learning (ICML)*.

Martens, J. & Sutskever, I. (2012) Training deep and recurrent networks with hessian-free optimization. *Neural Networks: Tricks of the Trade*. Berlin Heidelberg: Springer, pp. 479–535.

Mukherjee, I., Canini, K., Frongillo, R. & Singer, Y. (2013) Parallel boosting with momentum. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Berlin Heidelberg: Springer, pp. 17–32.

Nocedal, J. & Wright, S. (1999) *Numerical Optimization*, 2nd edn. New York, New York: Springer.

Pasupathy, R., Glynn, P., Ghosh, S. & Hashemi, F. S. (2015) On sampling rates in stochastic recursions. Under Review.

Pilanci, M. & Wainwright, M. J. (2015) Newton sketch: a linear-time optimization algorithm with linear-quadratic convergence. arXiv preprint arXiv:1505.02250.

Roosta–Khorasani, F. & Mahoney, M. W. (2016a) Sub-sampled Newton methods I: globally convergent algorithms. arXiv preprint arXiv:1601.04737.

Roosta–Khorasani, F. & Mahoney, M. W. (2016b) Sub-sampled Newton methods II: local convergence rates. arXiv preprint arXiv:1601.04738.

Tropp, J. A. & Wright, S. J. (2010) Computational methods for sparse solution of linear inverse problems. *Proc. IEEE*, **98**, 948–958.

Xu, P., Yang, J., Roosta–Khorasani, F., Ré, C. & Mahoney, M. W. (2016) *Sub-sampled newton methods with non-uniform sampling*. Advances in Neural Information Processing Systems, vol. 29. Red Hook, New York: Curran Associates, Inc. pp. 3000–3008.

Appendix

Additional numerical results

Numerical results on the rest of the datasets listed in Table 2 are presented here.

Note: The MNIST Dataset has been used for binary classification of digits into even and odd.

We only report results for the scaled covertype dataset, which is a very difficult problem when the data are unscaled. We were unable to tune the inner steplength in Newton-SGI to obtain reasonable performance for the unscaled dataset. (We suspect that the SGI iteration eventually converges, but extremely slowly.)

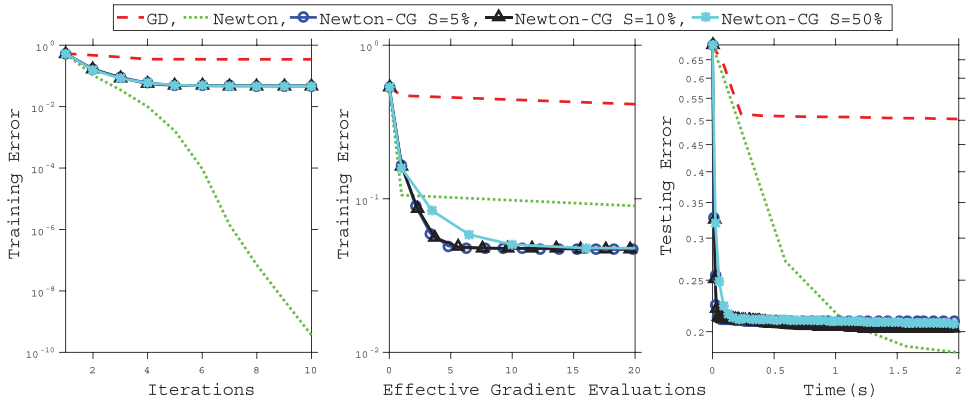

Fig. A1. **Cina Dataset**: Performance of the inexact subsampled Newton method (Newton-CG), using three values of the sample size, and of the GD and Newton methods. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Objective vs Time.

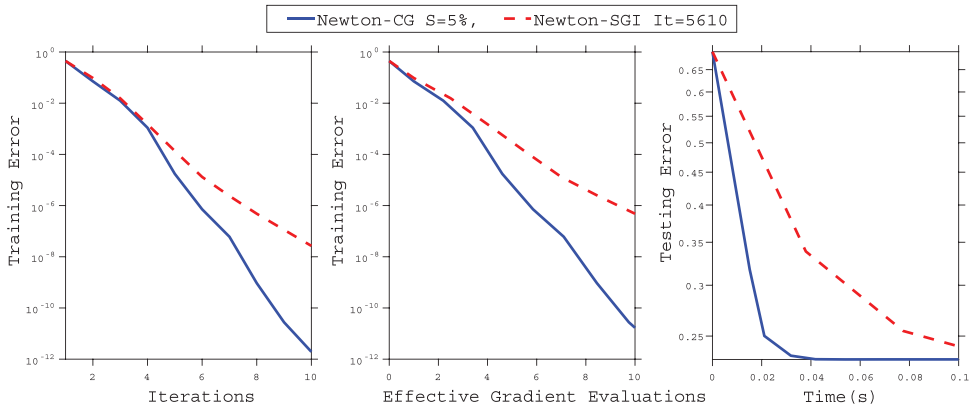

FIG. A2. **Cina Dataset (scaled)**: Comparison of Newton-CG with Newton-SGI. Left: Training Error vs Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Error vs. Time. The number of SGI iterations is determined through $It = |S|r$.

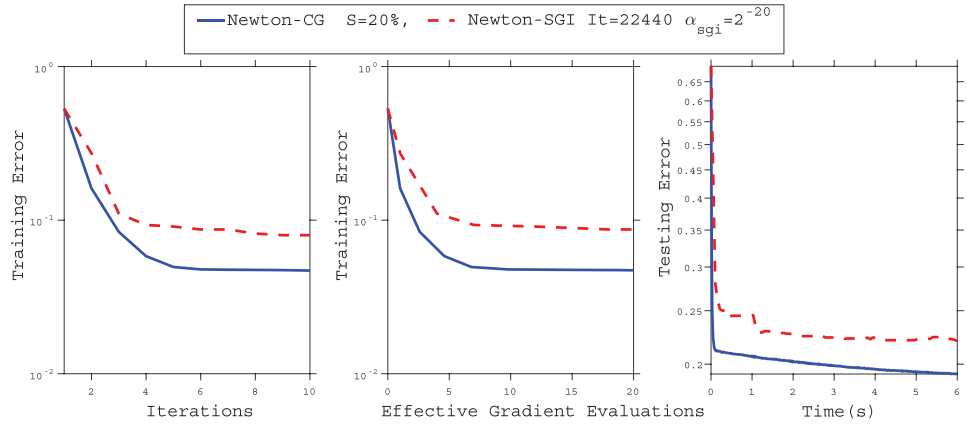

FIG. A3. **Cina Dataset (unscaled)**: Comparison of Newton-CG with Newton-SGI. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Objective vs. Time.

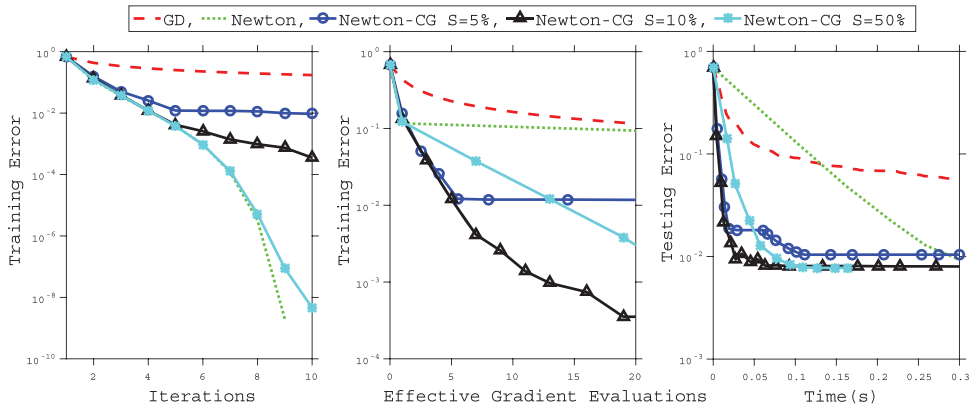

FIG. A4. **Mushrooms Dataset**: Performance of the inexact subsampled Newton method (Newton-CG), using three values of the sample size, against two other methods. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Objective vs. Time.

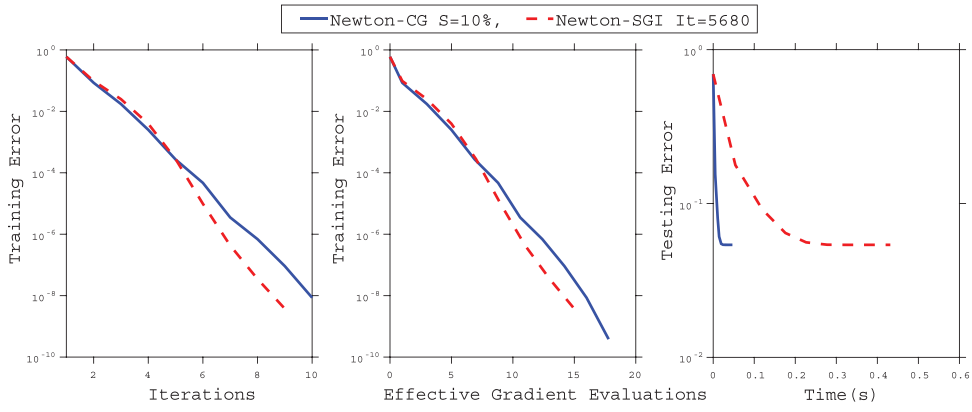

FIG. A5. **Mushrooms Dataset (scaled)**: Comparison of Newton-CG with Newton-SGI. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Error vs. Time.

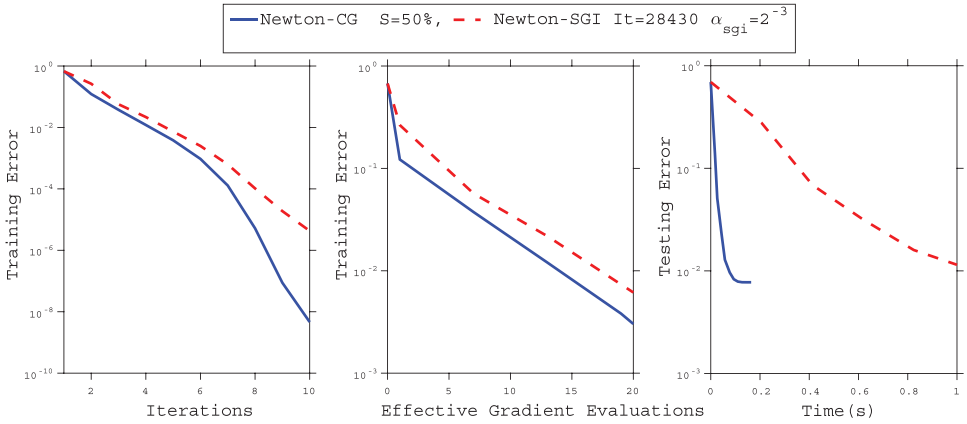

FIG. A6. **Mushrooms Dataset (unscaled)**: Comparison of Newton-CG with Newton-SGI. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Objective vs. Time.

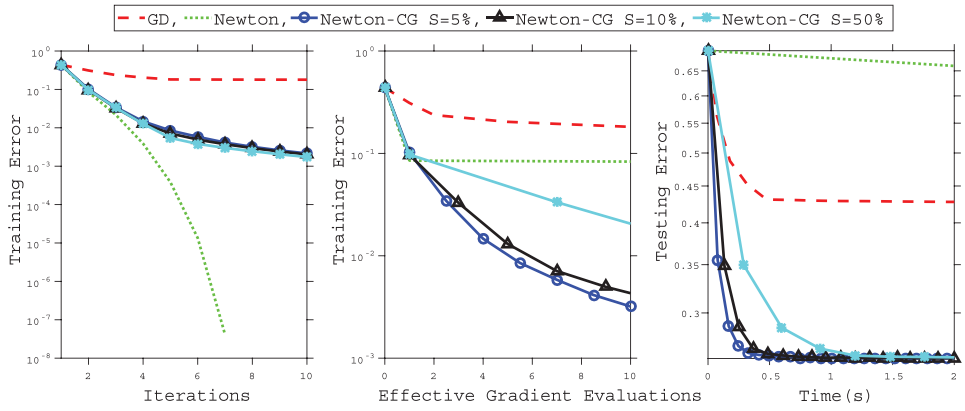

FIG. A7. **MNIST Dataset**: Performance of the inexact subsampled Newton method (Newton-CG), using three values of the sample size, and of the GD and Newton methods. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Objective vs. Time.

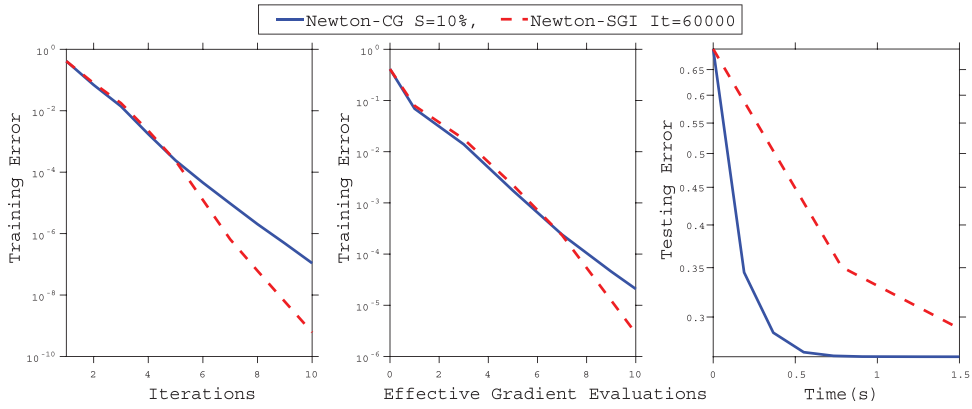

FIG. A8. **MNIST Dataset (scaled)**: Comparison of Newton-CG with Newton-SGI. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Error vs. Time.

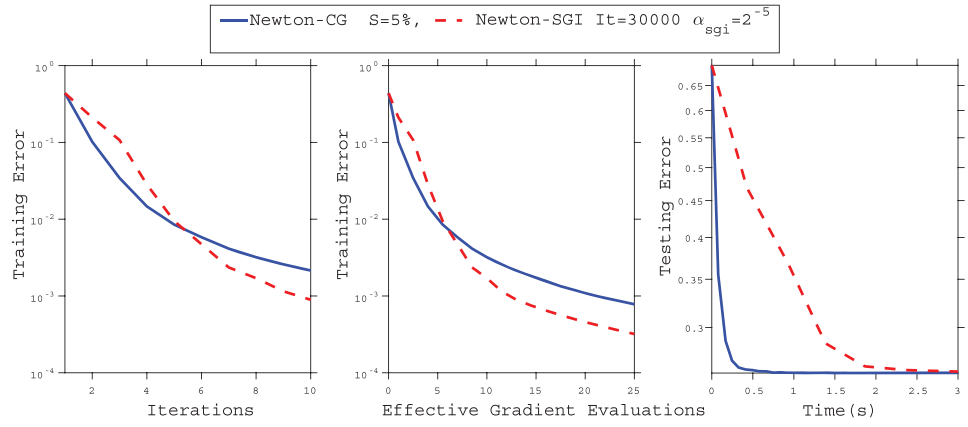

FIG. A9. **MNIST Dataset (unscaled)**: Comparison of Newton-CG with Newton-SGI. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Objective vs. Time.

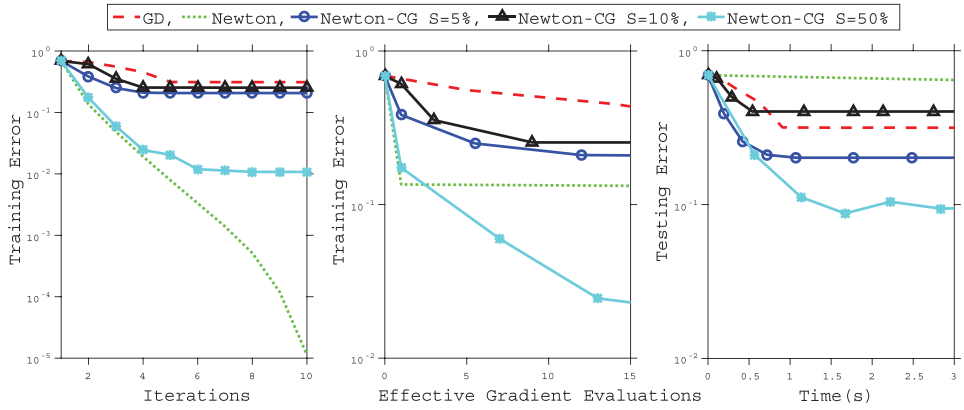

FIG. A10. **Gisette Dataset**: Performance of the inexact subsampled Newton method (Newton-CG), using three values of the sample size, and of the GD and Newton methods. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Objective vs. Time.

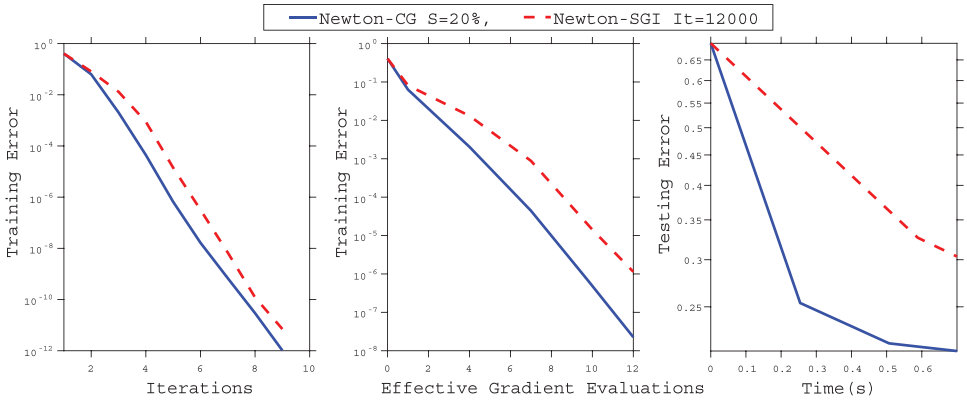

FIG. A11. **Gisette Dataset (scaled)**: Comparison of Newton-CG with Newton-SGI. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Error vs. Time.

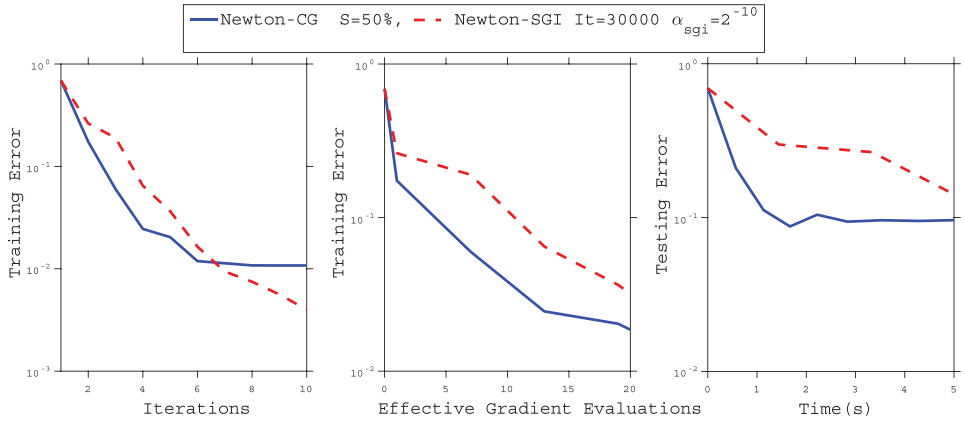

FIG. A12. **Gisette Dataset (unscaled)**: Comparison of Newton-CG with Newton-SGI. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Objective vs. Time.

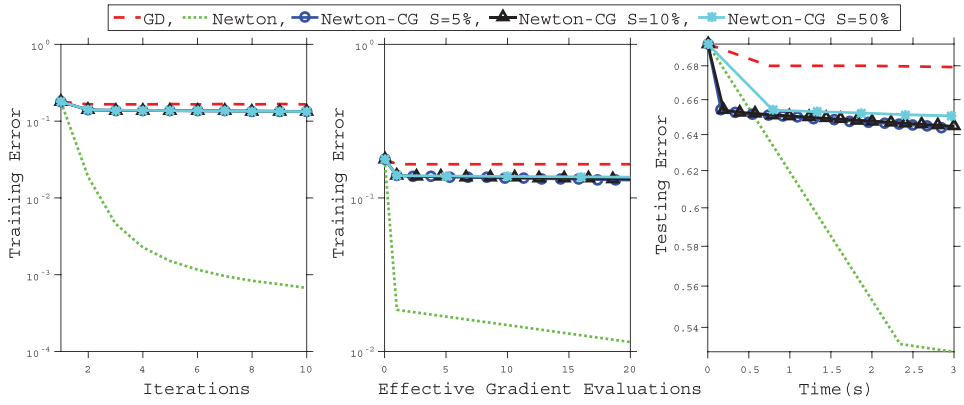

FIG. A13. **Covertype Dataset**: Performance of the inexact subsampled Newton method (Newton-CG), using three values of the sample size, and of the GD and Newton methods. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Objective vs. Time.

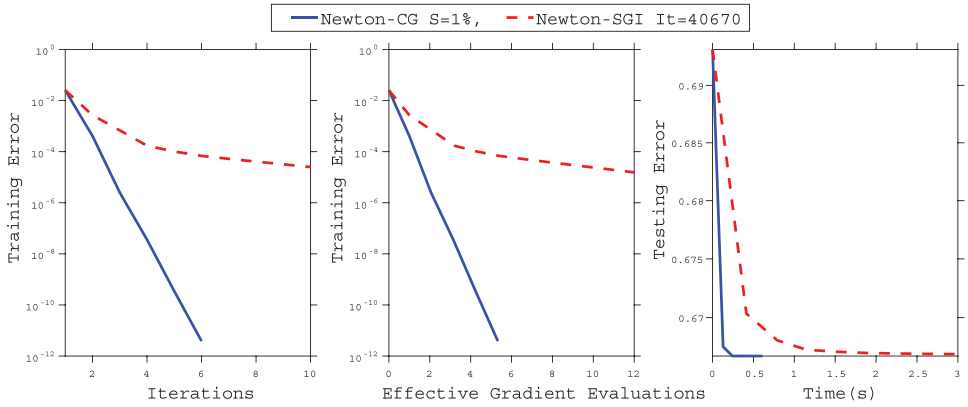

FIG. A14. **Covertype Dataset (scaled)**: Comparison of Newton-CG with Newton-SGI. Left: Training Error vs. Iterations; middle: Training Error vs. Effective Gradient Evaluations; right: Testing Error vs. Time.