Subsampled Inexact Newton methods for minimizing large sums of convex functions

Stefania Bellavia *  Nataša Krejić †  Nataša Krklec Jerinkić ‡

November 14, 2018

Abstract

This paper deals with the minimization of large sum of convex functions by Inexact Newton (IN) methods employing subsampled functions, gradients and Hessian approximations. The Conjugate Gradient method is used to compute the inexact Newton step and global convergence is enforced by a nonmonotone line search procedure. The aim is to obtain methods with affordable costs and fast convergence. Assuming strongly convex functions, R-linear convergence and worst-case iteration complexity of the procedure are investigated when functions and gradients are approximated with increasing accuracy. A set of rules for the forcing parameters and subsample Hessian sizes are derived that ensure local q-linear/superlinear convergence of the proposed method. The random choice of the Hessian subsample is also considered and convergence in the mean square, both for finite and infinite sums of functions, is proved. Finally, global convergence with asymptotic R-linear rate of IN methods is extended to the case of sum of convex function and strongly convex objective function. Numerical results on well known binary classification problems are also given. Adaptive strategies for selecting forcing terms and Hessian subsample size, streaming out of the theoretical analysis, are employed and the numerical results showed that they yield effective IN methods.

Key words: Inexact Newton, subsampled Hessian, superlinear convergence, global convergence, mean square convergence

*Department of Industrial Engineering, University of Florence, Viale Morgagni, 40/44, 50134 Florence, Italy, e-mail: stefania.bellavia@unifi.it. Research supported by Gruppo Nazionale per il Calcolo Scientifico, (GNCS-INdAM) of Italy.
†Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia, e-mail: natasa.krklec@dmi.uns.ac.rs. Research supported by Serbian Ministry of Education Science and Technological Development, grant no. 174030.
‡Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia, e-mail: natasa.krklec@dmi.uns.ac.rs. Research supported by Serbian Ministry of Education, Science and Technological Development, grant no. 174030.
1 Introduction

The problem we consider is

$$\min f_N(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

with $x \in \mathbb{R}^n$, $N$ - very large and all functions $f_i : \mathbb{R}^n \to \mathbb{R}$ convex and $f_N$ strongly convex. We are also interested in the case of large dimension $n$. There is a number of important problems of this type. To start with, one can be interested in minimizing the objective function stated in the form of mathematical expectation, $f(x) = E[F(x, w)]$, with $w$ being a random variable from some probability space. Given that the analytical expression for mathematical expectation is rarely available, one possibility is to approximate the expectation with Sample Average Approximation, SAA, function. In that case, a sample $\{w^1, \ldots, w^N\}$ is generated and the approximate objective function of the form (1) with $f_i(x) = F(x, w^i)$ is minimized. To ensure good approximation of the original objective function in general, one has to take a very large sample and thus calculating $f_N(x)$, its gradient and Hessian is expensive.

Binary and multi-class classification problems, e.g., employing softmax activation function and cross-entropy loss can also be expressed in the form (1). For a given set of data (usually very large), we are interested in classifying the data according to a set of rules specified by the data features. This problem has attracted a lot of attention recently due to its importance in machine learning. As the data set is generally very large it is imperative to use methods that can be implemented with reasonable costs.

In the framework of classical optimization, (1) is a convex problem that can be solved either by a first order or a second order method. However, the size of $N$ makes classical approaches prohibitively costly and thus calls for specific methods. One possibility is to consider different subsampling schemes which are used to reduce the cost of calculating $f_N$, its gradient and Hessian. There are many approaches in the literature, based on the idea of using a small sample subset at the beginning of iterative process and increasing the sample size as the solution is approached, ranging from simple heuristic approach [19, 9] to elaborate schemes that take into account the progress achieved up to the current iteration, [14, 31].

Whatever scheduling one adopts, it is assumed that the full sample is eventually reached, at least for the objective function and thus, the next question to be discussed is the choice of method. First order methods are attractive due to their low cost. One successful example is the stochastic gradient method, that employs a smaller subset of gradient components and thus reduce the cost even further, [19]. On the other hand several papers investigate the use of second order methods in this framework and demonstrate advantages in some important problems if the second order methods are correctly implemented, [4, 7, 5, 10, 11, 18, 26, 25, 34, 35, 36, 37, 40, 41]. For a comprehensive discussion of this issue one can see [8] and references cited therein.

In this paper we focus on subsampled Inexact Newton (IN) methods for (1) wrapped in a nonmonotone linesearch strategy. As it is well known, in Inexact (or Truncated) Newton methods [14, 31], the Newton equation is approximately solved and in case of large scale problems an iterative Krylov method is used to compute an approximate
solution of the Newton equation. The convexity of the objective function we are dealing with allows us to use CG method [23].

The choice of the nonmonotone strategy is motivated by the fact that the method uses approximate functions, at least initially, before the full sample is reached. Then, enforcing strict decrease in the Armijo rule might require unnecessary small steps. We adopt the nonmonotone line search procedure introduced in [28] and, assuming that each of the functions $f_i$ is strongly convex, we prove R-linear convergence. Also the worst-case iteration complexity is investigated and it is proved that the worst-case complexity bound of this class of non-monotone algorithms, analyzed in [20], is maintained provided that errors in gradient and function also decay with R-rate. Namely, the method requires at most $O(\log(\frac{1}{\varepsilon}))$ iterations to reach $f(x^k) - f(x^*) < \varepsilon$, where $x^*$ is the minimizer of (1).

Then, we turn our attention to the local properties of the method with the aim to obtain a local convergence rate faster than the R-linear convergence provided by first-order methods. The local convergence rate of IN methods with full Hessian depends on the choice of forcing terms which governs the error in solving each Newton linear system [16]. Here, as the Hessian of the objective function given in (1) might be prohibitively expensive to compute, we concentrate on subsampled Hessian and IN method that employs such Hessian approximations. We point out that in this context, it is pointless to solve the Newton equation exactly as the Hessian is generally approximated with a lower accuracy than function and gradient and the Newton model employed is actually a subsampled Newton model. Therefore, the use of CG method that allows us to control the accuracy in the solution of the Newton equation is advisable even if $n$ is not large [9].

Assuming that the sample size scheduling is given for the objective function and the gradient, i.e. assuming that eventually one reaches the full sample size $N$, we analyze the local convergence of a subsampled Hessian IN method. The analysis provides bounds on the Hessian accuracy requirements that depend on the employed forcing terms. Adaptive forcing terms, streaming out of the iterative process itself are derived as well. Furthermore, it is shown that the local method combines well with the nonmonotone line search, i.e., the $q$-linear/$q$-superlinear convergence rate of the local method are preserved.

In the second part of the paper we consider a randomized method obtained by relaxing the conditions for Hessian subsampling. Hence we prove the $q$-linear/superlinear convergence in the mean sense assuming that the Hessian approximation is good enough with high probability. The analysis yields relation between the Hessian subsample size, the forcing term and the (computable) sampled gradient at each iteration. Q-linear convergence in the mean sense is proved for fixed forcing terms with a fixed Hessian subsample size, while superlinear convergence in the mean square sense is obtained for the forcing terms that approach zero and increasing Hessian subsample sizes.

Having in mind the binary classification problem and the fact that the number of training points is enlarged over time in many applications, we also consider the case of unbounded $N$, i.e., the case where the objective function is defined as the mathematical expectation. For this problem we obtain convergence in the mean square sense as well.

Finally, the strong convexity assumption is relaxed similarly to the problem considered in [35, 37]. A bound on the Hessian sample size, derived in [41] is used to
obtain Hessians approximations positive definite with some high probability and CG is adapted to deal with possibly singular problems. The convergence in the the mean square is obtained for this problem as well.

From the performed theoretical analysis we derive adaptive rules for selecting both the forcing term sequence and the Hessian sample size. Particularly important feature of the proposed method is that the Hessian sample size is related to the current forcing term and approximated gradient norm, both quantities actually computable. Moreover, when $q$-superlinear convergence is sought, the Hessian sample size is adaptively chosen along the iterations, a low accuracy and smaller Hessian sample size are generally used in the early stage of the method while the accuracy and the Hessian sample size increase in the last stage of the convergence. Finally, we note that the Hessian sample size is also allowed to decrease if too high accuracy is used at the previous iterations. Numerical results on binary classification problems give numerical evidence of the proposed adaptive choices effectiveness.

The paper is organized as follows. In Section 2 the method is introduced and the global and local analysis is performed using the standard deterministic reasoning for the case of strongly convex functions $f_i$. Convergence in the mean square is proved in Section 3, considering all three cases - finite number of strongly convex functions $f_i$, an infinite number of strongly convex functions $f_i$’s and the last case with relaxed convexity assumptions. Some numerical results are presented in Section 4.

1.1 Related work

Our analysis is strictly related to that developed in [5, 7, 36, 37], where convergence of Inexact subsampled Newton methods is investigated both in probability [36, 37] and expectation [9, 7, 5]. We differ from these papers as we focus on the choice of the forcing terms, on the nonmonotone line search strategy and on adaptive choices of Hessian sample size.

Results presented in [7, 37] focus on the local behaviour of the method and give bounds for the accuracy required in the last stage of the procedure. The issue of developing an automatic transition between the initial stage of the procedure where a low accuracy in the Hessian approximation is enough, and the last stage where more accurate Hessian approximations are needed, is not investigated. However, in [37] the analysis is carried out under weaker assumptions than those we used here as the function $f_N$ is supposed to be strongly convex only in a neighborhood of the sought minimizer without any assumptions on convexity of functions $f_i$. A set of conditions on the gradient and the Hessian sample sizes that ensures local $R$-superlinear convergence in the expectation under the assumption on the variance of the error norms (Bounded moment of Iterates) is given in [7]. The Hessian sample size is assumed to be increased at each iteration starting from a large enough initial sample size.

In [9] an adaptive rule for choosing the gradient sample size is proposed along with an automatic criterion for the forcing term related to a variance estimation of the Hessian accuracy. The Hessian sample size is a fixed fraction of of the used gradient sample size.

Finally, we would like to mention that in [5] the authors perform a local complexity analysis of subsampled Inexact Newton methods and also show that methods that in-
corporate stochastic second-order information can be far more efficient on badly-scaled or ill-conditioned problems than first-order methods.

2 Inexact subsampled Newton method

We first introduce the notation and give some preliminary results. Throughout the paper \( \mathcal{N}_k \subset \{1, 2, \ldots, N\} \) denotes the sample used to approximate the objective function and its gradient, \( N_k \) denotes its cardinality and the subsampled function and gradient are defined as

\[
f_{N_k}(x) = \frac{1}{N_k} \sum_{i \in N_k} f_i(x), \quad \nabla f_{N_k} = \frac{1}{N_k} \sum_{i \in N_k} \nabla f_i(x).
\]

Moreover, \( D_k \subset \{1, 2, \ldots, N\} \) is the sample used for Hessian approximation with cardinality \( D_k \), and the subsampled Hessian is given by

\[
\nabla^2 f_{D_k}(x) = \frac{1}{D_k} \sum_{i \in D_k} \nabla^2 f_i(x).
\]

Here, we will consider subsampled Inexact Newton methods, that is iterative processes where at iteration \( k \), given the current iterate \( x^k \), the step \( s^k \) used to update the iterate satisfies

\[
\nabla^2 f_{D_k}(x^k)s^k = -\nabla f_{N_k}(x^k) + r^k, \quad \|r^k\| \leq \eta_k \|\nabla f_{N_k}(x^k)\|.
\]

The term \( \eta_k \) belongs to \((0, 1)\) and it is called forcing term \([14, 31]\). Here and in the rest of paper \( \| \cdot \| \) denotes the 2-norm.

Through the paper we will restrict our attention to convex functions, more precisely we will first consider strongly convex functions and then, in Subsection 3.2 relax the strong convexity to convexity. Let us state this formally.

**Assumption A1** The functions \( f_i, \ i = 1, \ldots, N \) are twice continuously differentiable and strongly convex, i.e.,

\[
\lambda_1 I \preceq \nabla^2 f_i(x) \preceq \lambda_n I, \quad \forall x \in \mathbb{R}^n \quad i = 1, \ldots, N
\]

with \( \lambda_1 > 0 \).

Assumption A1 implies a couple of inequalities that will be used further on. First of all, for all \( x \in \mathbb{R}^n \) we have

\[
\lambda_1 \|x - x^*\| \leq \|\nabla f_N(x)\| \leq \lambda_n \|x - x^*\|, \quad (5)
\]

where \( x^* \) is the unique minimizer of the function \( f_N \). Furthermore, according to \([32, \text{Theorem 2.10}]\), for every \( x \in \mathbb{R}^n \)

\[
\frac{\lambda_1}{2} \|x - x^*\|^2 \leq f_N(x) - f_N(x^*) \leq \frac{1}{\lambda_1} \|\nabla f_N(x)\|^2. \quad (6)
\]

Since \( \nabla^2 f_D(x) \) is positive definite we choose CG as the linear solver for computing \( s^k \) in \([3]\). Thus, we assume that CG initialized with zero vector is employed at each Inexact Newton iteration. We will make use of the following technical Lemma that it is proved in \([17]\).
Lemma 2.1 \[17\] Let $Ax = b$, where $A$ is a symmetric and positive definite matrix. Furthermore, let us assume that CG is applied to this system and it is terminated at the $i$th iteration. Then if CG is initialized with the null vector the approximate solution $x_i$ satisfies $x_i^TAx_i = x_i^Tb$.

Now, the above Lemma and Assumption A1 clearly imply the following result.

Lemma 2.2 Assume that $s^k$ satisfying (3) is obtained through CG method initialized with the null vector applied to the linear system

$$
\nabla^2 f_{D_k}(x^k)s = -\nabla f_{N_k}(x^k).
$$

Then, $\|s^k\| \leq \lambda_1^{-1}\|\nabla f_{N_k}(x^k)\|$.

2.1 Global convergence

In this section we analyze the behavior of the subsampled IN method and CG as the inner solver wrapped into the nonmonotone line-search strategy given in [28]. The choice of this line search rule is motivated by the fact that in the first stage of the procedure we work with subsampled functions and gradients and enforcing a strict Armijo type decrease might yield unnecessary small steps without real decrease in the original objective function. Thus, we allow additional freedom in the step size selection introducing in the Armijo condition a positive term usually denoted as error term, [28].

The line search procedure is governed by an error sequence $\{\nu_k\}$ with the following properties

$$
\nu_k > 0, \quad \sum_k \nu_k < \infty. \tag{7}
$$

Iteration $k$ of the above procedure, denoted as Algorithm GIN, is sketched in Algorithm 2.1.

**Algorithm 2.1: $k$-th iteration of Algorithm GIN**

Given $x^k \in \mathbb{R}^n$, $c \in (0, 1)$, $\bar{\eta} \in (0, 1)$ and $\{\nu_k\}$ such that (7) holds.

Step 1. Choose $N_k$, $D_k$, $\eta_k \in (0, \bar{\eta})$.

Step 2. Apply CG method initialized by the null vector to $\nabla^2 f_{D_k}(x^k)s^k = -\nabla f_{N_k}(x^k)$ and compute $s^k$ satisfying (3).

Step 3. Find the smallest nonnegative integer $j$ such that for $t_k = 2^{-j}$ there holds

$$
f_{N_k}(x^k + t_k s^k) \leq f_{N_k}(x^k) + ct_k(s^k)^T \nabla f_{N_k}(x^k) + \nu_k \tag{8}
$$

and set $x^{k+1} = x^k + t_k s^k$, $k = k + 1$. 

6
Then there exists \( t \) due to (4). Therefore we have proved for a schedule that eventually ends up with the full sample in a finite number of iterations. On the other hand, we have Assumption A1, the step size of algorithm GIN. Then (8) holds for \( t \) and thus, under Assumption A1, the step size \( t_k \) is strictly positive even for the standard Armijo rule in Step 3. The lower bound for \( t_k \) is obtained in Lemma 2.4 similarly to [20].

**Lemma 2.3** Let \( s^k \) be the step generated in Step 2 of Algorithm GIN. Then
\[
-(\nabla f_{N_k}(x^k))^T s^k \geq \frac{\lambda_1}{\lambda_n^2} (1 - \eta_k)^2 \|\nabla f_{N_k}(x^k)\|^2.
\]

**Proof.** The inexact condition implies
\[
\|\nabla f_{N_k}(x^k) - r_k\| \geq \|\nabla f_{N_k}(x^k)\| - \|r_k\| \geq (1 - \eta_k)\|\nabla f_{N_k}(x^k)\|.
\]

On the other hand, we have
\[
\|\nabla f_{N_k}(x^k) - r_k\| = \|\nabla^2 f_{D_k}(x^k) s^k\| \leq \|\nabla^2 f_{D_k}(x^k)\| \|s^k\| \leq \lambda_n \| s^k \|
\]
due to (4). Therefore
\[
\| s^k \| \geq \frac{\| \nabla f_{N_k}(x^k) - r_k \|}{\lambda_n} \geq \frac{1 - \eta_k}{\lambda_n} \| \nabla f_{N_k}(x^k) \|.
\]

(9)

Then, Lemma 2.1 and (4) yield
\[
-(\nabla f_{N_k}(x^k))^T s^k = (s^k)^T \nabla^2 f_{D_k}(x^k) s^k \geq \lambda_1 \| s^k \|^2 \geq \frac{\lambda_1}{\lambda_n^2} (1 - \eta_k)^2 \| \nabla f_{N_k}(x^k) \|^2.
\]

(10)

(11)

□

**Lemma 2.4** Suppose that the assumption A1 holds and let \( s^k \) be generated in Step 2 of algorithm GIN. Then (8) holds for \( t_k \geq \bar{t} = (1 - c)\lambda_1 / \lambda_n \).

**Proof.** Let \( k \) be an arbitrary iteration. If \( t_k = 1 \) satisfies (8), that is in Step 3 we have \( j = 0 \), then \( t_k \) is greater than \( \bar{t} \), as \( \bar{t} \in (0, 1) \). So let us consider the case \( t_k < 1 \). Then there exists \( t_k' = 2t_k \) such that
\[
f_{N_k}(x^k + t_k's^k) > f_{N_k}(x^k) + ct_k'(s^k)^T \nabla f_{N_k}(x^k) + \nu_k \geq f_{N_k}(x^k) + ct_k'(s^k)^T \nabla f_{N_k}(x^k).
\]
On the other hand, Assumption A1 implies, using the standard arguments for functions with bounded Hessians,

\[ f_{N_k}(x^k + t_k s^k) = f_{N_k}(x^k) + \int_0^1 (\nabla f_{N_k}(x^k + y t_k s^k))^T (t_k s^k) dy \]

\[ \leq \frac{\lambda_n}{2} (t_k')^2 \| s^k \|^2 + f_{N_k}(x^k) + t_k' (\nabla f_{N_k}(x^k))^T s^k. \]

Combining the previous two inequalities we obtain

\[ c t_k' (s^k)^T \nabla f_{N_k}(x^k) \leq \frac{\lambda_n}{2} (t_k')^2 \| s^k \|^2 + t_k' (\nabla f_{N_k}(x^k))^T s^k. \]

Dividing by \( t_k' \) and using \( t_k = t_k'/2 \), by rearranging the previous inequality we get

\[ t_k \geq \frac{(1 - c)(\nabla f_{N_k}(x^k))^T s^k}{\lambda_n \| s^k \|^2}. \] (12)

Now, the result follows from (10) and the fact

\[ \min \{ 1, (1 - c) \lambda_1 / \lambda_n \} = (1 - c) \lambda_1 / \lambda_n. \]

□

To prove the main results we need the following Lemma from [25].

**Lemma 2.5** Assume that \( \zeta_k \to 0 \) R-linearly. Then, for every \( \rho \in (0, 1) \),

\[ a_k = \sum_{j=1}^k \rho^{j-1} \zeta_{k-j} \]

converges to zero R-linearly.

Let us denote with \( \xi_k^f \) and \( \xi_k^g \) the inaccuracy in function and gradient, i.e. \( \xi_k^f \) and \( \xi_k^g \) are such that:

\[ \max_{x \in \{x_k, x_{k+1}\}} |f_{N_k}(x) - f_N(x)| \leq \xi_k^f, \quad \|\nabla f_{N_k}(x^k)\|^2 - \|\nabla f_N(x^k)\|^2 \leq \xi_k^g. \] (13)

Following Grapiglia and Sachs [20], we now prove that despite inaccuracy in function and gradient, Algorithm GIN meets the complexity results of nonmonotone line search methods with exact function and gradients.

**Theorem 2.1** Suppose that Assumption A1 holds and

\[ \sum_{k=0}^{\infty} \xi_k^f < \infty. \] (14)

Then, for a given \( \epsilon \in (0, 1) \) Algorithm GIN takes at most

\[ \bar{k} = \left[ \frac{f_N(x^0) - f_N(x^*) + \sum_{k=0}^{\infty} (2\xi_k^f + \nu_k) \epsilon^{-2}}{\kappa_c} \right], \]

iterations to ensure \( \|\nabla f_{N_k}(x^k)\| \leq \epsilon \), where

\[ \kappa_c = c(1 - c) \frac{\lambda_1^2}{\lambda_n^2 (1 - \bar{\eta})^2}. \] (15)
Proof. Note that, by Lemma 2.3 there follows
\[
(\nabla f_{N_k}(x^k))^T s^k \leq -\frac{\lambda_1}{\lambda_2 n^2} (1 - \bar{\eta})^2 \|\nabla f_{N_k}(x^k)\|^2. \tag{16}
\]
Then, we can proceed as in the proof of Theorem 1 in [20]. Let \( \bar{k} \) be the first iteration such that \( \|\nabla f_{N_k}(x^k)\| \leq \epsilon \). By (8) we obtain
\[
\nu_k + f_{N_k}(x^k) - f_{N_k}(x^{k+1}) \geq -ct_k (\nabla f_{N_k}(x^k))^T s^k, \quad k = 0, 1, \ldots, \bar{k} - 1.
\]
Moreover, by (16) and Lemma 2.4 there follows
\[
-ct_k (\nabla f_{N_k}(x^k))^T s^k \geq c(1 - c) \frac{\lambda_1^2}{\lambda_2 n^3} (1 - \bar{\eta})^2 \|\nabla f_{N_k}(x^k)\|^2.
\]
Then, for \( k = 0, 1, \ldots, \bar{k} - 1 \), there holds
\[
\nu_k + f_{N_k}(x^k) - f_{N_k}(x^{k+1}) \geq \kappa_c \epsilon^2
\]
with \( \kappa_c \) given in (15).
Therefore, by (13)
\[
\nu_k + 2\xi_k^f + f_{N}(x^k) - f_{N}(x^{k+1}) \geq \kappa_c \epsilon^2 \quad k = 0, \ldots, \bar{k} - 1.
\]
Summing up for \( k = 0, \ldots, \bar{k} - 1 \) we get
\[
\sum_{k=0}^{\infty} (2\xi_k^f + \nu_k) + f_{N}(x^0) - f_{N}(x^{\bar{k}}) \geq k \kappa_c \epsilon^2
\]
and the thesis follows. \( \Box \)

Note that in the previous theorem if \( N_k < N \) then
\[
\|\nabla f_{N_k}(x^k)\| \leq c_k + \xi_k^g. \tag{17}
\]

In what follows we prove \( R \)-linear convergence and consequently complexity estimates which are far better than \( O(\epsilon^{-2}) \) at the price of a strengthened assumption on the accuracy in functions and gradients.

**Theorem 2.2** Assume that A1 holds and let \( \{x^k\} \) be generated by Algorithm GIN. If the error sequences \( \{\nu_k\}, \{\xi_k^f\} \) and \( \{\xi_k^g\} \) converge to zero \( R \)-linearly then \( \{x^k\} \) converges \( R \)-linearly to the solution of (1).

**Proof.** Inequalities (8), (13) and (6) and Lemma 2.3 imply
\[
f_{N_k}(x^{k+1}) - f_{N}(x^*) \leq f_{N_k}(x^k) - f_{N}(x^*) + ct_k (\nabla f_{N_k}(x^k))^T s^k + \nu_k
\]
\[
\leq f_{N_k}(x^k) - f_{N}(x^*) - ct_k \frac{\lambda_1}{\lambda_2 n^2} (1 - \eta_k)^2 \|\nabla f_{N_k}(x^k)\|^2 + \nu_k
\]
\[
\leq f_{N_k}(x^k) - f_{N}(x^*) - ct_k \frac{\lambda_1}{\lambda_2 n^2} (1 - \eta_k)^2 (\lambda_1 (f_{N}(x^k) - f_{N}(x^*)) - \xi_k^g) + \nu_k
\]
Then, using Lemma 2.4 and (13) again, we obtain

$$f_N(x^{k+1}) - f_N(x^*) \leq \rho (f_N(x^k) - f_N(x^*)) + \bar{\xi}_k$$

(18)

where $\rho = 1 - c \frac{\lambda^2}{\lambda} (1 - \bar{\eta})^2 \in (0, 1)$ and $\bar{\xi}_k = \nu_k + 2\xi_k + \xi_k^g$. Furthermore, we obtain

$$f_N(x^{k+1}) - f_N(x^*) \leq \rho^{k+1} (f_N(x^0) - f_N(x^*)) + \rho \sum_{j=1}^{k} \rho^{j-1} \bar{\xi}_{k-j} + \bar{\xi}_k.$$  (19)

Thus, Lemma 2.5 yields the statement. □

Notice that R-linear convergence result obtained in Theorem 2.2 also holds for $\nu_k = 0$, that is, for the Armijo line search. The theorem above also allows us to prove the complexity result below.

Theorem 2.3 Assume that A1 holds. If the error sequences $\{\nu_k\}, \{\xi^f_k\}$ and $\{\xi^g_k\}$ converge to zero R-linearly then for any $\epsilon \in (0, \epsilon^{-1})$, there exist $\tilde{\rho} \in (0, 1)$ and $Q > 0$ such that Algorithm GIN takes at most

$$\tilde{k} = \left\lfloor \frac{\log((f_N(x^0) - f_N(x^*) + Q))}{\log(\tilde{\rho})} \right\rfloor \log(\epsilon^{-1})$$

iterations to ensure $f_N(x^{\tilde{k}}) - f_N(x^*) < \epsilon$.

Proof. Assumptions of Theorem 2.2 are satisfied and

$$f_N(x^k) - f_N(x^*) \leq \rho^k (f_N(x^0) - f_N(x^*)) + \sum_{j=0}^{k-1} \rho^j \bar{\xi}_{k-j}.$$  (20)

Given that the sequence $\left\{\sum_{j=0}^{k} \rho^j \bar{\xi}_{k-j}\right\}$ converges to zero R-linearly by Lemma 2.5 there exist $\bar{\rho} \in (0, 1)$ and $Q > 0$ such that

$$\sum_{j=0}^{k-1} \rho^j \bar{\xi}_{k-j} \leq Q \bar{\rho}^k.$$  

Therefore, for $\bar{\rho} = \max\{\rho, \bar{\rho}\}$, we have from (20)

$$f_N(x^k) - f_N(x^*) \leq \bar{\rho}^k (f_N(x^0) - f_N(x^*) + Q)$$

and the statement follows as in Theorem 6 of [20]. □

A couple of comment is due here. First of all, Theorems 2.1-2.3 deal with the possibility of infinite sequence of errors in the objective function and the gradient. So, these statements provide a framework for considering unbounded $N$ as well. However, our focus here is on the problems with fixed $N$. Thus, any kind of scheduling for $N_k$ that ensures the existence of $k_0$ such that $N_k = N$ for $k \geq k_0$ actually implies that the sequences $\{\xi^f_k\}$ and $\{\xi^g_k\}$ are finite, with $k$ nonzero elements at most. Therefore, the statements of the above theorems apply to any kind of scheduling that ensures reaching
the full sample for the objective function. Theorem 2.3 proves the complexity bound of \( \log(\epsilon^{-1}) \), although we work with cheaper objective function and the gradient whenever \( N_k < N \). Thus the theorem above provides theoretical justification for working with smaller samples. Moreover, the analysis carried out so far does not involve either the forcing term or the accuracy in Hessian approximation. Then, the results we provided hold even if \( D_k = 1 \) and only one CG-iteration is performed at each iteration of Algorithm GIN.

### 2.2 Local convergence

In this section we assume that the scheduling of the sample sizes \( N_k \) is given and that eventually we reach \( f_N \) and \( \nabla f_N \) at some iteration \( \tilde{k} \) and continue with \( N_k = N \) for \( k \geq \tilde{k} \). Then, we may restrict all asymptotic theoretical considerations to the method defined with \( N_k = N \), that is the step \( s^k \) satisfies

\[
\nabla^2 f_{D_k}(x^k)s^k = -\nabla f_N(x^k) + r^k, \quad \|r^k\| \leq \eta_k \|\nabla f_N(x^k)\|.
\]

(21)

Following the analysis in [16], in this subsection we focus on the local \( q \)-linear and \( q \)-superlinear convergence of method GIN. Then, we assume that the generated sequence converges to the solution of (1), relying on the global convergence results proved in the previous subsection. The analysis presented here differs from that in [16] as we have to take into account the approximation in the Hessian and enlightens that the accuracy in the Hessian’s approximation must be related to the adopted forcing term. This theoretical study enables us to devise an adaptive and computable rule for the Hessian sample size yielding \( q \)-linear and \( q \)-superlinear convergence. In fact, this is what we should obtain in order to justify the use of a second order method. The analysis is carried out under the following assumption on the subsampled Hessian matrices.

**Assumption A2.** There exists a constant \( L \) such that for any \( \mathcal{D} \subseteq \{1, 2, \ldots, N\} \) we have

\[
\|\nabla^2 f_D(x) - \nabla^2 f_D(x^*)\| \leq L\|x - x^*\|,
\]

(22)

for \( x \) sufficiently close to \( x^* \).

The above Assumption holds if there exists a neighborhood of \( x^* \) where each Hessian \( \nabla^2 f_i \) is Lipschitz continuous. Note also that Assumption A1 implies that all function \( \nabla f_i \) are Lipschitz continuous with the constant \( \lambda_n \).

Moreover, we assume that the error in the Hessian approximation is determined only by the subsample size, independently of the sample taken and denote by \( h(D, x) \) the norm of the error in the Hessian approximation for a given subsample size \( D \) at point \( x \), i.e.

\[
h(D, x) := \|\nabla^2 f_N(x) - \nabla^2 f_D(x)\|.
\]

We make the following assumption on the subsample size \( D_k \) used at each iteration \( k \).

**Assumption A3.** At each iteration \( k \) of Algorithm GIN the subsample size \( D_k \) is chosen such that

\[
h(D_k, x^k) \leq C\eta_k,
\]

(23)

with \( 0 < C < (1/\bar{\eta} - 1)\lambda_1/2 \).
In the subsequent analysis for any $\delta > 0$ the ball with center $x^*$ and radius $\delta$ will be denoted by $N_\delta(x^*)$, i.e. $N_\delta(x^*) = \{ x \in \mathbb{R}^n : \| x - x^* \| \leq \delta \}$. Moreover, we let $\delta^*$ sufficiently small such that (22) holds for any $x$ in $N_{\delta^*}(x^*)$ and $\delta^* < 2 \min\{\lambda_1, 1/\lambda_n\}/L$.

The next Theorem states that the full step is taken in Algorithm GIN eventually.

**Theorem 2.4** Assume that the sequence $\{x^k\}$ generated by Algorithm GIN converges to $x^*$. Let Assumptions A1 and A3 hold and $c \in (0, 1/4)$ in (8). Then, there exists $k_0$ such that for all $k \geq k_0$ the full step $s^k$ is accepted in Step 3 of Algorithm GIN.

**Proof.** Take $\bar{\epsilon} \in (0, \bar{\eta}/2)$ and $\bar{\delta} > 0$ such that

$$\| \nabla f_N(x^k + \xi s^k) - \nabla^2 f_N(x^k) \| \leq \bar{\epsilon}, \text{ for all } \| s^k \| \leq \bar{\delta}, \xi \in (0, 1).$$

(24)

According to Lemma 2.2, $\lim_{k \to \infty} \| s^k \| = 0$. Then, there exists $k_0$ such that (24) holds for $k \geq k_0$. Moreover, the Taylor expansion yields

$$f_N(x^k + s^k) = f_N(x^k) + (\nabla f_N(x^k))^T s^k + \frac{1}{2}(s^k)^T \nabla^2 f_N(\theta^k)s^k$$

$$= f_N(x^k) + (\nabla f_N(x^k))^T s^k + \frac{1}{2}(s^k)^T \nabla^2 f_{D_k}(x^k)s^k$$

$$+ \frac{1}{2}(s^k)^T (\nabla^2 f_N(\theta^k) - \nabla^2 f_N(x^k))s^k + \frac{1}{2}(s^k)^T (\nabla^2 f_N(x^k) - \nabla^2 f_{D_k}(x^k))s^k;$$

(25)

with $\theta^k = x^k + \xi s^k$, $\xi \in (0, 1)$. From Assumption A3 we obtain

$$(s^k)^T (\nabla^2 f_N(x^k) - \nabla^2 f_{D_k}(x^k))s^k \leq h(D_k, x^k)\| s^k \|^2 \leq C\eta_k\| s^k \|^2 \leq C\bar{\eta}\| s^k \|^2.$$  

(26)

Recall that Lemma 2.1 implies

$$(s^k)^T \nabla^2 f_{D_k}(x^k)s^k = -\nabla f_N(x^k)^T s^k.$$  

(27)

Putting (24), (26) and (27) into (25) we get

$$f_N(x^k + s^k) \leq f_N(x^k) + \frac{1}{2}(\nabla f_N(x^k))^T s^k + \frac{1}{2}(\bar{\epsilon} + C\bar{\eta})\| s^k \|^2.$$  

(28)

From (10), we conclude that

$$f_N(x^k + s^k) \leq f_N(x^k) + \frac{1}{2}(1 - \frac{\bar{\epsilon} + C\bar{\eta}}{\lambda_1})(\nabla f_N(x^k))^T s^k.$$  

Note that from $C < (1/\bar{\eta} - 1)/\lambda_1$ there follows $\frac{C\bar{\eta}}{\lambda_1} < (1 - \bar{\eta})/2$. Therefore, the choice of $\bar{\epsilon}$ yields $\bar{\epsilon}/\lambda_1 + C\bar{\eta}/\lambda_1 < 1/2$. Then,

$$f_N(x^k + s^k) \leq f_N(x^k) + \frac{1}{4}\nabla f_N^T(x^k)s^k$$

and condition (8) is satisfied with $t_k = 1$ for any $k > k_0$ as $c \in (0, 1/4)$. □

The following Lemma, whose proofs can be found in the Appendix, is needed in the subsequent convergence analysis.
Lemma 2.6 Let Assumptions A1-A2 hold. If \( x^k \in N_{\delta^*}(x^*) \), \( s^k \in \mathbb{R}^n \) and \( \eta \in (0, 1) \) are such that \( \| \nabla f_N(x^k) + \nabla^2 f_{D_k}(x^k)s^k \| \leq \eta \| \nabla f_N(x^k) \| \) and \( x^k + s^k \in N_{\delta^*}(x^*) \) then
\[
\| \nabla f_N(x^k + s^k) \| \leq (\eta + B(x^k)) \| \nabla f_N(x^k) \|,
\]
with \( B(x^k) = \frac{1}{\lambda_1^2} L \| \nabla f_N(x^k) \| + h(D, x^k) \).

Lemma 2.7 Let Assumption A1 holds and \( \delta \in (0, \delta^*/(1 + \lambda_1^{-1} \lambda_n)) \). If \( x^k \in N_{\delta}(x^*) \) and \( \| \nabla f_N(x^k) + \nabla^2 f_{D_k}(x^k)s^k \| \leq \| \nabla f_N(x^k) \| \) then \( x^k + s^k \in N_{\delta^*}(x^*) \).

The convergence of \( \{ x^k \} \) together with (5), implies the following result.

Theorem 2.5 Assume that the sequence \( \{ x^k \} \) generated by Algorithm GIN converges to \( x^* \). Let Assumptions A1-A3 hold and \( c \in (0, 1/4) \) in (3). If \( \eta_k = \bar{\eta} \) at each iteration of Algorithm GIN, then the sequence \( \{ x^k \} \) converges to \( x^* \)-linearly for \( \bar{\eta} \) small enough. Moreover, if \( \lim_{k \to \infty} \eta_k = 0 \), the convergence is \( q \)-superlinear.

Proof. Note that, by the choice of \( C \) in Assumption A3, \( \bar{\eta}(1 + \lambda_1^{-1} C) < (1 + \eta)/2 \). Let \( \delta \in (0, \delta^*/(1 + \lambda_1^{-1} \lambda_n)) \). Take \( \bar{\varepsilon} \in (0, \delta \lambda_1] \) sufficiently small such that \( \bar{\eta}(1 + \lambda_1^{-1} C) + \lambda_1^{-2}L\bar{\varepsilon}/2 < \tau < 1 \) for some \( \tau \in (0, 1) \). Let \( k_0 \) be defined as in Theorem 2.4 and \( \bar{k} \geq k_0 \), such that \( x^k \in N_{\delta}(x^*) \) sufficiently near to \( x^* \) to guarantee \( \| \nabla f_N(x^k) \| \leq \bar{\varepsilon} \).

Lemma 2.7 yields \( x^{k+1} \in N_{\delta^*}(x^*) \) and by Lemma 2.6 we obtain
\[
\| \nabla f_N(x^{k+1}) \| \leq \tau \| \nabla f_N(x^k) \| \leq \| \nabla f_N(x^k) \| \leq \bar{\varepsilon}.
\]
Therefore, using (5),
\[
\| x^{k+1} - x^* \| \leq \frac{1}{\lambda_1} \| \nabla f_N(x^{k+1}) \| \leq \frac{1}{\lambda_1} \bar{\varepsilon} \leq \delta,
\]
so \( x^{k+1} \in N_{\delta}(x^*) \) and \( \| \nabla f_N(x^{k+1}) \| \leq \bar{\varepsilon} \).

As an inductive hypothesis suppose that for some \( k \geq \bar{k} \) we have \( x^k \in N_{\delta}(x^*) \) and \( \| \nabla f_N(x^k) \| \leq \bar{\varepsilon} \). Then \( x^{k+1} = x^k + s^k \in N_{\delta^*}(x^*) \) by Lemma 2.7 and Lemma 2.6 implies
\[
\| \nabla f_N(x^{k+1}) \| \leq \frac{1}{\lambda_1} \| \nabla f_N(x^k) \| \leq \| \nabla f_N(x^k) \| \leq \bar{\varepsilon}.
\]
Again, (5) yields \( \| x^{k+1} - x^* \| \leq \delta \) and \( x^{k+1} \in N_{\delta}(x^*) \). Therefore, proceeding by induction we conclude that \( x^k \in N_{\delta^*}(x^*) \) for any \( k \geq \bar{k} \) and by Lemma 2.6
\[
\| \nabla f_N(x^{k+1}) \| \leq \frac{1}{\lambda_1} \| \nabla f_N(x^k) \| \leq \| \nabla f_N(x^k) \| \leq \bar{\varepsilon}.
\]
(29)
Therefore, as \( \| \nabla f_N(x^k) \| \to 0 \), using (5) we obtain that \( \{ x^k \} \) converges to \( x^* \) with \( q \)-linear rate provided that
\[
\bar{\eta} < \frac{\lambda_1}{\lambda_n} \frac{1 + \lambda_1^{-1} C}{1 + \lambda_1^{-2} L}. \]
Moreover, the $q$-superlinear convergence follows if $\lim_{k \to \infty} \eta_k = 0$. □

The above results are in line with the classical convergence theory of Inexact Newton methods [14] as the local linear convergence require $\eta_k \leq \bar{\eta} < 1$ and the upper bound on $\bar{\eta}$ depends on inverse of the conditioning of the Hessian.

Let us now discuss one possible choice of $\eta_k$ in order to obtain $q$-superlinear convergence of the procedure. Following ideas in [16] our choice of $\eta_k$ depends on the agreement between the function and the subsampled Newton model. If there is a good agreement between these two quantities, even if the quality of the approximation in the Hessian is lower than that in the function, it is reasonable to use a small $\eta$ in the subsequent iteration. Let us consider the following choice of $\eta_k$ in Algorithm 2.1,

$$\eta_k = \min\left\{\bar{\eta}, \frac{|f_{N_k}(x^k) - m_{k-1}(s^{k-1})|}{\|\nabla f_{N_{k-1}}(x^{k-1})\|}\right\}, \bar{\eta} < 1$$  \hspace{1cm} (30)

where

$$m_{k-1}(s) = f_{N_{k-1}}(x^{k-1}) + \nabla f_{N_{k-1}}(x^{k-1})^T s + \frac{1}{2} s^T \nabla^2 f_{D_{k-1}}(x^{k-1}) s.$$  \hspace{1cm} (31)

In the following theorem we show that the sequence $\{\eta_k\}$ generated by (30) converges to zero and ensures $q$-superlinear convergence of the sequence generated by Algorithm GIN if the scheduling ensures that the full sample is reached at some finite iteration.

**Theorem 2.6** Let assumptions in Theorem 2.5 hold and $\eta_k$ given by (30). Then $\{x^k\}$ converges to $x^*$ superlinearly.

**Proof.** Let the iteration index $k > \tilde{k} + 1$, i.e. such that $N_{k-1} = N_k = \mathcal{N}$. Using the Taylor’s expansion and (23) we obtain

$$|f_N(x^k) - m_{k-1}(s^{k-1})| \leq \frac{1}{2} \|s^{k-1}\|^2 \left(\frac{L}{2} \|s^{k-1}\| + C \eta_{k-1}\right).$$  \hspace{1cm} (32)

Now, by Lemma 2.2 there follows

$$\eta_k \leq \frac{1}{2} \lambda_1^{-2} \|\nabla f_N(x^{k-1})\|^2 \left[\lambda_1^{-1} \frac{L}{2} \|\nabla f_N(x^{k-1})\| + C \bar{\eta}\right].$$  \hspace{1cm} (33)

Then, as $\lim_{k \to \infty} \|\nabla f_N(x^k)\| = 0$ we have $\lim_{k \to \infty} \eta_k = 0$ and the superlinear convergence follows by Theorem 2.5. □

The above result can be proved also choosing $\eta_k$ as

$$\eta_k = \min\left\{\bar{\eta}, \frac{|f_{N_k}(x^k) - m_{k-1}(s^{k-1})|}{\omega_k}\right\}, \bar{\eta} < 1$$  \hspace{1cm} (34)

with

$$\omega_k = O(\|\nabla f_{N_{k-1}}(x^{k-1})\|).$$

We also underline that Theorem 2.6 in case of full Hessian, i.e. $N_k = D_k$, shows that the Truncated Newton method for unconstrained optimization problems with the choice of forcing terms given by (34) is superlinearly convergent. Such result also follows from the analysis presented in [27, Theorem 3.10].
3 Mean square convergence

Large part of the previous analysis strongly relies on the Hessian error bound \( h(D, x) \). However, assessing this quantity is not an easy task in general. In this section, we provide some more specific guidance on choosing \( D_k \). We assume that the subset \( D_k \) is chosen randomly and we relax controlling \( h(D, x) \) in such way that we ask for a good enough Hessian approximation with some probability smaller than 1. Thus less conservative estimates are now feasible. More precisely, we use a bound similar to that derived in [41] to carry out our analysis. Consequently, we obtain stochastic convergence results - convergence in a mean square sense (m.s.). Our main result is that the \( q \)-linear convergence in m.s. can be achieved with a small enough but fixed forcing term \( \eta \) and with large enough but fixed Hessian sample size \( D \). On the other hand, to achieve \( q \)-superlinear convergence in m.s. with \( \eta_k \) defined by (34), the Hessian sample size is required to increase as \( \eta_k \) goes to zero. This analysis paths the way to devise adaptive rules for selecting the Hessian sample size in order to use a small Hessian sample size in the early stage of the procedure, when \( \eta_k \) is close to one and linear systems are solved only to a low accuracy, and automatically increase it when the solution is approached.

In this section we assume that the subsample \( D \) is chosen randomly and uniformly - every \( \nabla^2 f_i(x) \) has the same chance to be chosen. Let \( D \) be any subset of \( N \) such that \(|D| = D\). Then one can derive a bound on \( D \) such that the following holds for a given \( \gamma > 0 \) and \( \alpha \in (0, 1) \)

\[
P(\|\nabla^2 f_D(x) - \nabla^2 f_N(x)\| \leq \gamma) \geq 1 - \alpha.
\]  

(35)

The corresponding bound is stated in Lemma 3.1 (see the proof in the Appendix), while a similar bound is provided in Lemma 4 of [41]. The result is obtained by using the Bernstein inequality, see [39] and [30] for further references.

Lemma 3.1 Assume that A1 holds and that the subsample \( D \) is chosen randomly and uniformly from \( N \). Let \( \gamma > 0 \) and \( \alpha \in (0, 1) \) be given. Then (35) holds at any point \( x \) if the subsample size \( D \) satisfies

\[
D \geq \frac{2(\ln(2n/\alpha))}{\gamma^2} \left( \lambda_n^2 + \lambda_n \gamma/3 \right) := \bar{l}.
\]  

(36)

We use the above results and the analysis of the previous section, to design a globally convergent inexact subsampled Newton method with adaptive choice of the Hessian sample size. In particular we will choose \( D_k \) such that the inequality

\[
h(D_k, x^k) \leq C \max\{\eta_k, \|\nabla f_{N_k}(x^k)\|\}
\]  

(37)

holds with probability \( 1 - \alpha_k \), with \( \alpha_k \in (0, 1) \) and \( 0 < C < (1/\bar{\eta} - 1)\lambda_1/2 \). This corresponds to \( \gamma = C \max\{\eta_k, \|\nabla f_{N_k}(x^k)\|\} \) in (35).

3.1 Bounded sample - GIN-R method

In this subsection we are interested in the case of finite \( N \). Let us assume that the full sample is eventually reached for the objective function and the gradient, i.e., \( N_k = N \),
for \( k \) sufficiently large, say \( k \geq \bar{k} \). The procedure we obtain is based on GIN method but with a specific choice of the Hessian subsample \( D_k \), namely its cardinality is set according to (37) and the sample is randomly chosen. We denote this procedure as GIN-R to emphasize the specific random choice of the Hessian subsample and for the sake of convenience we list its generic iteration \( k \) in Algorithm 3.1, where we denote the first steps as Step 1.a-1.c to make clear that they correspond to specific choices in Step 1 of Algorithm GIN.

We will analyze the convergence in the m.s. considering two possibilities. If the forcing terms converge to zero, i.e., if \( \eta_k \to 0 \), then \( \gamma_k \) given by (38) converges to zero. The other case we consider is \( \eta_k = \bar{\eta} < 1 \). In that case Algorithm GIN-R yields \( \gamma_k \) bounded away from zero and we have \( D_k \subset N \) with \( D_k < N \) during the whole iterative process. Theorem 2.2 implies the R-linear convergence for any Hessian subsampling under the condition \( N_k = N', k \geq \bar{k} \). The key issue in the global convergence analysis is that we have descent search direction with an arbitrary good or poor Hessian approximation, i.e., regardless of the subsample used in GIN algorithm. The line search globalization strategy makes the algorithm convergent. Therefore, no matter how the subsample is chosen, the R-linear convergence still holds. Thus we immediately have the following statement.

**Algorithm 3.1: \( k \)-th iteration of Method GIN-R**

Given \( x^k \in \mathbb{R}^n, \bar{\eta} \in (0,1), c \in (0,1), C > 0, \nu_k, \alpha_k \in (0,1) \).

Step 1.a Choose \( N_k, \eta_k \in (0,\bar{\eta}) \).

Step 1.b Compute

\[
\gamma_k = C \max\{\eta_k, \|\nabla f_{N_k}(x^k)\|\}. \tag{38}
\]

Step 1.c Set \( \alpha = \alpha_k \) and \( \gamma = \gamma_k \) and compute \( D \) such that (36) holds. If \( D \geq N_k \) set \( D_k = N_k \). Else, choose the sample \( D_k \) randomly and uniformly from \( N_k \) such that \( D_k \geq D \).

Step 2 Apply CG method initialized by the null vector to

\[
\nabla^2 f_{D_k}(x^k)s^k = -\nabla f_{N_k}(x^k)
\]

and compute \( s^k \) satisfying (3). Set \( t_k = 2^{-j} \) and set \( x^{k+1} = x^k + t_k s^k \).

**Theorem 3.1** Assume that A1 holds and let \( \{x^k\} \) be generated by Algorithm GIN-R. If \( \{\nu_k\} \) converges to zero R-linearly then \( \{x^k\} \) converges R-linearly to \( x^* \).

Next we show another important intermediate result.
Lemma 3.2 Suppose that the assumptions of Theorem 3.1 are satisfied and let \( \{x^k\} \) be a sequence generated by Algorithm GIN-R. If \( \eta_k \) is defined by (30) then there exist positive constants \( B_1 \) and \( B_2 \) such that

\[
\eta_k \leq B_1 \tau^k \quad \text{and} \quad \gamma_k \leq B_2 \tau^k
\]

for all \( k \) large enough. If \( \eta_k = \bar{\eta} \) then \( \gamma_k = C \bar{\eta} \) for all \( k \) large enough.

Proof. First, Theorem 3.1 implies that the sequence of iterates \( x^k \) converges to the unique solutions \( R \)-linearly - for all \( k \) large enough we have

\[
\|x^k - x^*\| \leq B \tau^k
\]

(39)

where \( B > 0 \) and \( \tau \in (0, 1) \). Now, using the Taylor’s expansion, Lemma 2.2 and (4) we obtain for some \( \theta_k \in [x^{k-1}, x^k] \)

\[
| f_N(x^k) - m_{k-1}(s^{k-1}) | \\
= | f_N(x^{k-1}) + t_k \nabla f_N(x^{k-1})^T s^{k-1} + \frac{1}{2} t_k^2 (s^{k-1})^T \nabla^2 f_N(\theta_k) s^{k-1} \\
- f_N(x^{k-1}) - \nabla f_N(x^{k-1})^T s^{k-1} - \frac{1}{2} (s^{k-1})^T \nabla^2 f_{\partial_k}(x^{k-1}) s^{k-1} | \\
\leq \frac{1 - t_k}{\lambda_1} \| \nabla f_N(x^{k-1}) \|^2 + \lambda_n \| s^{k-1} \|^2 \\
\leq \frac{1}{\lambda_1} \| \nabla f_N(x^{k-1}) \|^2 + \frac{\lambda_n}{\lambda_1^2} \| \nabla f_N(x^{k-1}) \|^2
\]

Therefore, using (5), for \( \eta_k \) given by (30) we obtain

\[
\eta_k \leq \frac{| f_N(x^k) - m_{k-1}(s^{k-1}) |}{\| \nabla f_N(x^{k-1}) \|} \leq \left( \frac{1}{\lambda_1} + \lambda_n \| x^{k-1} - x^* \| \right) \leq B_1 \tau^k
\]

(40)

where \( B_1 = \left( \frac{1}{\lambda_1} + \lambda_n B / \tau \right) \). Moreover, for \( \gamma_k \) given by (38) we get

\[
\gamma_k \leq C \max\{ B_1 \tau^k, \lambda_n B \tau^k \} = C \max\{ B_1, \lambda_n B \} \tau^k := B_2 \tau^k.
\]

Considering the case with \( \eta_k = \bar{\eta} \), since the gradient converges to zero, for all \( k \) large enough we have \( \| \nabla f_N(x^k) \| < \bar{\eta} \) and therefore \( \gamma_k = C \bar{\eta} \). □

Now, we are ready to show the main result of this subsection.

Theorem 3.2 Suppose that the assumptions of Theorem 3.1 and Assumption A2 are satisfied. Moreover, assume \( c \in (0, 1/4) \) in (8) and \( 0 < C < (1/\bar{\eta} - 1) \lambda_1 / 2 \) in (37). Let \( \{x^k\} \) be a sequence generated by Algorithm GIN-R. Then, there are positive constants \( V_1, V_2, C_1, C_2 \) and \( \tau \in (0, 1) \) such that for all \( k \) sufficiently large

1. if \( \eta_k \) is defined by (30) then

\[
E(\| x^{k+1} - x^* \|^2) \leq \left( V_1 \tau^{2k} + V_2 \alpha_k \right) E(\| x^k - x^* \|^2);
\]

2. if \( \eta_k \) is defined by (35) then

\[
E(\| x^{k+1} - x^* \|^2) \leq \left( V_1 \tau^{2k} + V_2 \alpha_k \right) E(\| x^k - x^* \|^2);
\]

3. if \( \eta_k \) is defined by (36) then

\[
E(\| x^{k+1} - x^* \|^2) \leq \left( V_1 \tau^{2k} + V_2 \right) E(\| x^k - x^* \|^2);
\]

4. if \( \eta_k \) is defined by (37) then

\[
E(\| x^{k+1} - x^* \|^2) \leq \left( V_1 \tau^{2k} + V_2 \right) E(\| x^k - x^* \|^2);
\]
b) if \( \eta_k = \bar{\eta} \) is sufficiently small then

\[
E(\|x^{k+1} - x^*\|^2) \leq ((C_1 r^k + C_2 \bar{\eta})^2 + V_2 \alpha_k) E(\|x^k - x^*\|^2).
\]

**Proof.** Since the assumptions of Theorem 3.1 are satisfied, the sequence \( \{x^k\} \) converges to the solution \( R \)-linearly, independently of \( D_k \) and for \( \eta_k \in (0, 1) \). Thus, there exist constants \( B > 0 \) and \( \tau \in (0, 1) \) such that \( \|x^k - x^*\| \leq Br^k \). Moreover, \( N_k = N \) for all \( k \) sufficiently large. So, without loss of generality, we assume that the full sample is used for the gradient and the function and \( \|\nabla f_N(x^k)\| < \bar{\eta} \).

Employing (6) and Lemma 2.2 we have the following estimate

\[
\|x^k + t_k s^k - x^*\|^2 \leq \frac{2}{\lambda_1} (f_N(x^k) + t_k s^k) - f_N(x^*)
\]

\[
\leq \frac{2}{\lambda_1} (f_N(x^k) - f_N(x^*) + t_k \nabla f_N(x^k)^T s^k + \frac{1}{2\lambda_1} \|t_k s^k\|^2)
\]

\[
\leq \frac{2}{\lambda_1} \left( \frac{1}{\lambda_1} \|\nabla f_N(x^k)\|^2 + \frac{1}{2\lambda_1} \|\nabla f_N(x^k)\|^2 \right)
\]

\[
\leq \frac{2}{\lambda_1} \left( \frac{1}{\lambda_1} + \frac{1}{2\lambda_1} \right) \lambda_1^2 \|x^k - x^*\|^2 := V_2 \|x^k - x^*\|^2.
\]

Let us denote by \( A_k \) the event \( \|\nabla^2 f_{D_k}(x^k) - \nabla^2 f_{N}(x^k)\| \leq \gamma_k \). Due to Step 1.c) of algorithm GIN-R it follows that \( P(A_k) \geq 1 - \alpha_k \), i.e., \( P(\bar{A}_k) \leq \alpha_k \). Notice that (41) holds in both cases but in the case of \( A_k \) we can derive better estimate.

Assume that \( A_k \) happens. Then

\[
\|\nabla^2 f_N(x^k) s^k + \nabla f_N(x^k)\|
\]

\[
\leq \|\nabla^2 f_{D_k}(x^k) s^k + \nabla f_{D_k}(x^k)\| + \|\nabla^2 f_{D_k}(x^k) s^k + \nabla f_{N}(x^k)\|
\]

\[
\leq \gamma_k \|s^k\| + \eta_k \|\nabla f_N(x^k)\| \leq \gamma_k (\|x^k - x^*\| + \eta_k \|\nabla f_N(x^k)\|).
\]

Moreover, as \( \gamma_k < C\sqrt{\bar{\eta}} \) we can repeat the reasoning used in the proof of Theorem 2.4 to conclude that the full step is accepted for \( k \) sufficiently large. Also, notice that the standard assumptions for (Inexact) Newton method are satisfied, so following the standard steps for IN analysis one can prove

\[
\|x^k + s^k - x^*\| \leq c_1 (\|x^k - x^*\| + \gamma_k / \lambda_1 + \eta_k) \|x^k - x^*\|
\]

for some positive constant \( c_1 \) and for \( k \) sufficiently large. Indeed, denoting \( \tilde{r}_k = \nabla^2 f_N(x^k) s^k + \nabla f_N(x^k) \), it holds \( \|\tilde{r}_k\| \leq \gamma_k / \lambda_1 + \eta_k \|\nabla f_N(x^k)\| \) and

\[
\|x^k + s^k - x^*\| = \|x^k + (\nabla^2 f_N(x^k))^{-1} \tilde{r}_k - (\nabla^2 f_N(x^k))^{-1} \nabla f_N(x^k) - x^*\|.
\]

Since the Newton’s method converges quadratically there exists \( \kappa > 0 \) such that

\[
\|x^k - (\nabla^2 f_N(x^k))^{-1} \nabla f_N(x^k) - x^*\| \leq \kappa \|x^k - x^*\|^2.
\]

Now, using \( \| (\nabla^2 f_N(x^k))^{-1} \| \leq 1 / \lambda_1 \), \( \|\nabla f_N(x^k)\| \leq \lambda_n \|x^k - x^*\| \) and defining \( c_1 = \max\{\kappa, \lambda_n / \lambda_1\} \) we obtain (42). Using inequality (39) and squaring inequality (42) we obtain

\[
\|x^k + s^k - x^*\|^2 \leq 2c_1^2 (B^2 r_{2k} + (\gamma_k / \lambda_1 + \eta_k))^2 \|x^k - x^*\|^2.
\]
Now, we distinguish two cases depending on $\eta_k$. Using the result of Lemma 3.2 and assuming that $k$ is sufficiently large we obtain the following.

a) If $\eta_k$ is defined by (30), for $V_1 = 2c_1^2(B^2 + 2B_1^2 + 2(B_2/\lambda_1)^2)$ we get

$$E(\|x^{k+1} - x^*\|^2) = P(A_k)E(\|x^{k+1} - x^*\|^2\mid A_k) + P(\bar{A}_k)E(\|x^{k+1} - x^*\|^2\mid \bar{A}_k) \leq (V_1\tau^{2k} + \alpha_k V_2)E(\|x^k - x^*\|^2).$$

(44)

b) Considering $\eta_k = \bar{\eta}$, for $C_1 = 2c_1^2B^2$ and $C_2 = 2c_1^2(C/\lambda_1 + 1)^2$ we get

$$E(\|x^{k+1} - x^*\|^2) \leq (C_1\tau^{2k} + C_2\bar{\eta}^2 + V_2\alpha_k)E(\|x^k - x^*\|^2).$$

(45)

□

We conclude the analysis with the following corollary.

**Corollary 3.1** Assume that the conditions of Theorem 3.2 hold and let $\{x_k\}$ be a sequence generated with Algorithm GIN-R. Then the sequence $\{x_k\}$ converges to $x^*$ in the mean square sense:

a) linearly if $\eta_k = \bar{\eta}$ is sufficiently small and $\alpha_k < \frac{1}{\lambda_2} - \frac{C_2}{\bar{\eta}}$;

b) superlinearly if $\eta_k$ is defined by (30) and $\lim_{k \to \infty} \alpha_k = 0$.

Note that the requirement on the probability $\alpha_k$ in order to get the $q$-linear convergence is not too demanding in what concerns the sample size, because it influences only the logarithmic factor in (36).

### 3.2 Unbounded sample - GIN method

In many applications, the number of training points is enlarged over time so the cardinality of the sample set $N$ is actually unbounded. This motivated us to consider the following problem as well

$$\min_{x \in \mathbb{R}^n} f(x) = E(F(x, \xi)), \quad (46)$$

where $\xi$ is a random variable defined on a probability space $(A, \mathcal{F}, P)$ and $F$ is twice differentiable function with respect to $x$. Let us denote $f_i(x) := F(x, \xi_i)$ where $\xi_i, i = 1, 2, \ldots$ is an i.i.d. sequence of variables following the same distribution as $\xi$. For example, $\xi_i$ can represent the pair of input-output variables in machine learning problems. Then, we can use the same notation as in the previous sections to define SAA approximation of the objective function and its derivatives: $f_N, \nabla f_N, \nabla^2 f_N$. We will prove that, under appropriate assumptions, GIN converges in m.s. towards the solution of problem (46).

Let us formalize the assumption about random variables $\xi_i$.

**Assumption B1** $\xi_i, i = 1, 2, \ldots$ is an i.i.d. sequence of variables.

Next we assume that the sequence of iterates $\{x^k\}$ belongs to a bounded set. This assumption might sound like a strong one, but it is needed for convergence theory and
the analysis of properties of $F(x, \xi)$ that guaranty this property is beyond the scope of this paper. Notice that a weaker assumption is used in [7] (the assumption of bounded moments of iterates). However, here we are using a linesear search procedure to compute the steplength $t_k$ and therefore we need to use the SAA approximation for the function. On the contrary, [7] employs a fixed steplength whose definition require the knowledge of the maximum eigenvalue $\lambda_n$, but allows to get rid of computing the objective function.

**Assumption B2** There exists a compact set $\Omega$ such that $\{x^k\}_{k\in\mathbb{N}} \subseteq \Omega$.

**Assumption B3** $F(\cdot, \xi) \in C^2(\mathbb{R}^n)$ for every $\xi$. $F$ and $\nabla F$ are dominated by an integrable functions $M_F(\xi)$ and $M_\nu(\xi)$, respectively, on an open set containing $\Omega$.

Assumption B1 implies that $E(f_N(x)) = f(x)$. Moreover, B1 and B3 imply that $\nabla f(x) = E(\nabla F(x, \xi))$, and therefore $E(\nabla f_N(x)) = \nabla f(x)$. Furthermore, the Uniform law of large numbers implies that $f_N$ and $\nabla f_N$ almost surely (a.s.) converge to $f(x)$ and $\nabla f(x)$, respectively, uniformly on $\Omega$ when $N$ tends to infinity. Let us write this down more formally. Denote

$$e_N = \max_{x \in \Omega} |f_N(x) - f(x)|, \quad \hat{e}_N = \max_{x \in \Omega} ||\nabla f_N(x) - \nabla f(x)||.$$  \hspace{1cm} (47)

Then, $\lim_{N \to \infty} e_N = 0$ and $\lim_{N \to \infty} \hat{e}_N = 0$, a.s. and using the Lebesgue Dominated Convergence Theorem (see Theorem 7.31 of [38]) we obtain

$$\lim_{N \to \infty} E(e_N) = 0, \quad \lim_{N \to \infty} E(\hat{e}_N) = 0.$$ \hspace{1cm} (48)

Assuming the strong convexity of $f_i$ as in assumption A1, it is easy to show that $f_N$ is also strongly convex with the same constants $\lambda_1$ and $\lambda_n$ for any $N$. Moreover, assuming B3, $f$ also remains strongly convex with the constant $\lambda_1$. Indeed, for an arbitrary $i$ and $x, y$ there holds

$$f_i(y) \geq f_i(x) + \nabla^T f_i(x)(y - x) + \frac{\lambda_1}{2} ||x - y||^2.$$  

Taking the expectation and using that $E(\nabla f_i(x)) = \nabla f(x)$ we obtain the strong convexity of $f$. Therefore, problem (46) has an unique solution $x^*$.

**Theorem 3.3** Suppose that the assumptions A1, B1-B3 hold and that $N_k \to \infty$. Then any sequence $\{x^k\}_{k \in \mathbb{N}}$ generated by GIN converges towards the solution of the problem (46) in the mean square sense.

**Proof.** First, notice that assumptions B1-B3 imply that $|f_N(x^k) - f(x^k)| \leq e_N$ for every $k$, so following the reasoning as in the proof of Theorem 2.2 we obtain

$$f(x^{k+1}) - f(x^*) \leq f_N(x^k) - \tilde{q}^2 ||\nabla f_N(x^k)||^2 + e_{N_k} - f(x^*) + \nu_k,$$ \hspace{1cm} (49)

where $\tilde{q} = (1 - c)\lambda_1/\lambda_n$ and $q = (\lambda_1(1 - \eta)^2)/(\lambda_n^2)$. Now,

\[
\begin{align*}
||\nabla f_N(x^k)||^2 &= ||\nabla f_N(x^k) - \nabla f(x^k) + \nabla f(x^k)||^2 \\
&= ||\nabla f(x^k)||^2 + 2(\nabla f(x^k))^T(\nabla f_N(x^k) - \nabla f(x^k)) + ||\nabla f_N(x^k) - \nabla f(x^k)||^2 \\
&\geq ||\nabla f(x^k)||^2 - 2||\nabla f(x^k)||||\nabla f_N(x^k) - \nabla f(x^k)|| \\
&\geq ||\nabla f(x^k)||^2 - 2M_\nu e_{N_k},
\end{align*}
\hspace{1cm} (50)
\]
where the last inequality follows from (47), continuity of $\nabla f$, Assumption B2 and $M_g = \max_{x \in \Omega} \|\nabla f(x)\|$. On the other hand, strong convexity of $f$ implies that

$$-\|\nabla f(x_k)\|^2 \leq -\lambda_1 (f(x_k) - f(x^*)) \leq (f(x_k) - f(x^*)) (1 - \omega) + 2e_{N_k}^* + 2c\bar{t}qM_g\tilde{e}_{N_k} + \nu_k,$$

where $\omega = c\bar{t}q\lambda_1 \in (0, 1)$. Applying expectation we get

$$E(f(x_k+1) - f(x^*)) \leq E(f(x_k) - f(x^*)) (1 - \omega) + a_k,$$

where $a_k = 2E(e_{N_k}^*) + 2c\bar{t}qM_gE(\tilde{e}_{N_k}) + \nu_k$. Now, (48), (7) and the assumption that $N_k \to \infty$ together imply that $\lim_{k \to \infty} a_k = 0$. Therefore, it follows (see [22]) that

$$\lim_{k \to \infty} E(f(x_k) - f(x^*)) = 0.$$

Finally, strong convexity implies $|x_k - x^*|^2 \leq (f(x_k) - f(x^*)) / \lambda_1$ thus

$$\lim_{k \to \infty} E(|x_k - x^*|^2) = 0$$

\[ \square \]

### 3.3 Relaxing the strong convexity - method GIN-M

In this part of the paper, we consider a relaxation of the strong convexity assumption by letting $\nabla^2 f_i(x)$ be only positive semidefinite while the final objective function remains strongly convex. Similar assumptions are stated in [36]. Notice that Theorem 3.3 does not impose any assumption on the size of the Hessian subsample. But relaxation of the strong convexity imposes condition on the Hessian sample size needed to obtain a positive definite Hessian in probability. We use the bound provided in [36] but the approach presented here differs in several ways from the one in [36]. First of all, we prove mean square convergence to the solution of (46) under appropriate conditions given in the sequel, while in [36] convergence with some (high) probability for finite sum problems like (1) is considered. Second, we continue to use CG as the inner solver - although modified in this case to cope with possibly singular matrix.

**Assumption C1** The functions $f$ and $F(\cdot, \xi)$ are twice continuously differentiable and there exist $0 < \lambda_1 < 1$ and $\lambda_n > 0$ such that for every $x, \xi$

$$0 \leq \nabla^2 F(x, \xi) \leq \lambda_n I \quad \text{and} \quad \lambda_1 I \leq \nabla^2 f(x) \leq \lambda_n I.$$

This assumption ensures that the unique solution of the original problem still exists. Moreover, we assume that the Hessian approximations are unbiased.

**Assumption C2** For every $x \in \mathbb{R}^n$ and every $i \in \mathbb{N}$ there holds $E(\nabla^2 f_i(x)) = \nabla^2 f(x)$.

This assumption allows us to use the Matrix Chernoff result (see [39]) and to obtain the bound presented in Lemma 1 of [36]. Although we observe unbounded sample, the same result holds. More precisely, under the assumptions C1 and C2 we obtain that, given $\mu \in (0, 1 - \lambda_1)$ the following holds
Let $V$ belong to the Krylov subspace $\| \cdot \|$. Then, noting that its iteration $k$ above, is used. We refer to the obtained procedure as Algorithm GINR-M and sketch its iteration $k$ in Algorithm 3.2.

\[ P(\lambda_{\min}(\nabla^2 f_D(x)) \geq \mu) \geq 1 - \alpha \]

if
\[ D \geq \frac{2\lambda_n(1 - \lambda_1)^2 \ln(n/\alpha)}{\mu^2 \lambda_1} := \bar{D}(\alpha). \]  

(51)

Given that the subsampled Hessian $\nabla^2 f_{D\ell}$ might be singular, for the computation of the step $s^k$ at Step 2 of Algorithm GIN-R, we proceed as follows. If at iteration $j$ of CG it happens that $(s^k_j)^T \nabla^2 f_{D\ell}(x^k)s^k_j = 0$ we stop CG and set $s^k = s^k_{j-1}$. Notice that $s^k$ is still a descent direction as $(s^k_{j-1})^T \nabla f = -(s^k_{j-1})^T \nabla^2 f_{D\ell}(x^k)s^k_{j-1} < 0$. The algorithm we use is again GIN-R where the sample size is selected such that subsampled Hessian is positive definite with probability $1 - \alpha$. The modified CG, as explained above, is used. We refer to the obtained procedure as Algorithm GINR-M and sketch its iteration $k$ in Algorithm 3.2.

\textbf{Algorithm 3.2: $k$-th iteration of Method GINR-M}

Given $x^k \in \mathbb{R}^n$, $c \in (0, 1), \tilde{\eta} \in (0, 1)$, $C > 0$, $\{\nu_k\}$, $\alpha \in (0, 1)$ $\mu \in (0, 1 - \lambda_1)$.

**Step 1.a** Choose $N_k, \eta_k \in (0, \tilde{\eta})$.

**Step 1.b** If $N_k \leq \bar{D}(\alpha)$ given in (51), set $D_k = N_k$. Else, choose $D_k \geq \bar{D}(\alpha)$ and the subsample $D_k$ randomly and uniformly from $N_k$.

**Step 2.** Determine $s^k$ with modified CG: if $(s^k_j)^T \nabla^2 f_{D\ell}(x^k)s^k_j = 0$ for some inner iteration $j$, set $s^k = s^k_{j-1}$. Otherwise, find the step $s^k$ such that (3) holds.

**Step 3.** Find the smallest nonnegative integer $j$ such that (3) holds for $t_k = 2^{-j}$ and set $x^{k+1} = x^k + t_k s^k$.

Relaxing the strong convexity results in losing the usual relation between the step and the gradient stated in Lemma 2.2. Therefore, we need the following assumption.

**Assumption C3** There exists a constant $M_s > 0$ such that the step generated by GINR-M satisfies $\|s^k\| \leq M_s$ for every $k$.

A comment is due with respect to the above Assumption. Assume that the step $s^k$ computed at Step 4 of GINR-M has been generated at iteration $j$ of CG. Then it belongs to the Krylov subspace

\[ \mathcal{K}_j = \text{span}\{\nabla f_{N_k}(x^k), (\nabla^2 f_{D\ell}(x^k))\nabla f_{N_k}(x^k), \ldots, (\nabla^2 f_{D\ell}(x^k))^{j-1} \nabla f_{N_k}(x^k)\}. \]

Let $V$ be an orthonormal basis of $\mathcal{K}_j$, then $s^k = V y$, where $y \in \mathbb{R}^j$. Therefore

\[ (s^k)^T \nabla^2 f_{D\ell}(x^k)s^k = y^T V^T \nabla^2 f_{D\ell}(x^k)V y \geq \lambda_{\min}(V^T \nabla^2 f_{D\ell}(x^k)V)\|y\|^2. \]

Then, noting that $\|s\| = \|y\|$ as $V$ is orthonormal, from

\[ (s^k)^T \nabla^2 f_{D\ell}(x^k)s^k = -(s^k)^T \nabla f_{N_k}(x^k) \]
Theorem 3.4 holds. by GINR-M converges towards the solution of (46) in m.s. infinity. Then there exist A

Denote by $A_k$ enough such that $\lambda_{\min}(\nabla^2 f_{D_k}(x^k)) \geq \mu$. Since $D_k$ satisfies (51) it follows that $P(\bar{A}_k) \leq \alpha$.

Assume that $\bar{A}_k$ happens. Then, we can proceed as in the proof of Theorem 3.3 as $f$ is strongly convex and $\lambda_{\min}(\nabla^2 f_{D_k}(x^k)) \geq \mu$ yields $(\nabla f_{N_k}(x^k))^T s^k \leq -\mu \|s^k\|^2$. We obtain

$$f(x^{k+1}) - f(x^*) \leq (f(x^k) - f(x^*)) (1 - \omega) + 2\epsilon N_k + \theta \dot{e}_{N_k} + \nu_k,$$

where $\omega = c\tilde{q}\lambda_1 \in (0, 1)$, $\tilde{t} = (1 - c)\mu/\lambda_n$, $q = \mu(1 - \tilde{t})^2$, $\theta = 2c\tilde{q}M_g$ and $M_g = \max_{x \in \Omega} \|\nabla f(x)\|$. On the other hand, assume that $\bar{A}_k$ happens. Then by the Taylor expansion and assumption C1 we obtain

$$f(x^{k+1}) - f(x^*) \leq f(x^k) - f(x^*) + t_k(\nabla f(x^k))^T s^k + \frac{1}{2} \lambda_n \|t_k s^k\|^2. \quad (52)$$

Again, using the strong convexity of $f$ we get

$$\|t_k s^k\|^2 = \|x^{k+1} - x^k\|^2 \leq 2(\|x^{k+1} - x^*\|^2 + \|x^k - x^*\|^2)$$

$$\leq \frac{4}{\lambda_1}(f(x^{k+1}) - f(x^*) + f(x^k) - f(x^*)). \quad (53)$$

Moreover,

$$(\nabla f(x^k))^T s^k = (s^k)^T (\nabla f(x^k) + \nabla f_{N_k}(x^k))$$

$$\leq \|s^k\| \|\nabla f(x^k) - \nabla f_{N_k}(x^k)\| + (\nabla f_{N_k}(x^k))^T s^k$$

$$= \|s^k\| \|\nabla f(x^k) - \nabla f_{N_k}(x^k)\| - (s^k)^T \nabla^2 f_{D_k}(x^k)s^k$$

$$\leq M_s \dot{e}_{N_k}, \quad (54)$$

where the last inequality follows from the fact that $\nabla^2 f_{D_k}(x^k)$ is positive semidefinite. Putting (54) into (52), together with (55) we obtain

$$f(x^{k+1}) - f(x^*) \leq (f(x^k) - f(x^*)) (1 + 2\lambda_n/\lambda_1) + 2\lambda_n/\lambda_1 (f(x^{k+1}) - f(x^*)) + M_s \dot{e}_{N_k}.$$
Combining all together we get
\[
E(f(x^{k+1}) - f(x^*)) = P(A_k)E(f(x^{k+1}) - f(x^*))|A_k) + P(\tilde{A}_k)E(f(x^{k+1}) - f(x^*)|\tilde{A}_k)
\]
\[
\leq E(f(x^k) - f(x^*)) (1 - \omega) + 2\varepsilon_{\mathcal{N}_k} + \theta E(\tilde{\varepsilon}_{\mathcal{N}_k}) + \nu_k + \alpha (E(f(x^k) - f(x^*)) (1 + 2\lambda_n/\lambda_1) + 2\lambda_n/\lambda_1 E(f(x^{k+1}) - f(x^*)))
\]
\[
+ \alpha M_s E(\tilde{\varepsilon}_{\mathcal{N}_k}).
\]

Rearranging the previous inequality and assuming \(\alpha < \frac{1}{2\lambda_n}\) we obtain
\[
E(f(x^{k+1}) - f(x^*)) \leq \tau E(f(x^k) - f(x^*)) + a_k,
\]
where
\[
\tau = 1 - \omega + \frac{\alpha (1 + 2\lambda_n/\lambda_1)}{u}, \quad u = 1 - \alpha 2\lambda_n/\lambda_1
\]
and
\[
a_k = \frac{1}{u} (\nu_k + 2\varepsilon_{\mathcal{N}_k} + (\theta + \alpha M_s) E(\tilde{\varepsilon}_{\mathcal{N}_k})).
\]
Notice that \(\tau \in (0, 1)\) provided that \(\alpha\) is small enough. Moreover, as discussed in the previous proof, \(a_k\) tends to zero and the result follows. \(\Box\)

4 Numerical results

In this section we report on our numerical experience with subsampled inexact Newton approaches. Our experiments were performed in Matlab R2017a, on a Intel Core i5-6600K CPU 3.50 GHz x 4 16GB RAM. For the approximate solution of the arising linear systems we used the conjugate gradient method (CG) implemented in the Matlab function \(\text{pcg}\). We did not employ a preconditioner and we used the conjugate gradient method in a matrix-free manner. Then, only the product of \(\nabla f^2_{D_k}\) with vectors is needed. The aim of this section it to provide numerical evidence of the benefits deriving by the employment of adaptive rules, streaming out from our theory, for choosing forcing terms and Hessian sample size. Then, in our numerical experimentation we use full gradients and functions, i.e. \(N_k = N\) for \(k > 0\) and we compare the full Inexact Newton (FIN) method with \(\eta = 10^{-4}\), the subsampled Hessian (SIN) method with \(\eta_k = 10^{-4}\) and \(D_k = 0.3N\) for all \(k\)'s, the subsampled inexact method with adaptive choices of \(\eta_k\)'s and \(D_k = 0.3N\) for all \(k\)'s (SINA_FT) and the subsampled inexact method with adaptive choices of \(\eta_k\)'s and \(D_k\)'s (SINA_FT_Dk). We also consider a subsampled method with constant \(D_k = 0.3N\) and a maximum number of five iterations allowed to \(\text{pcg}(\text{SIN_cg5})\).

In SINA_FT and SINA_FT_Dk, \(\eta_k\) is chosen as follows:
\[
\left\{
\begin{array}{c}
\eta_k = \min\{0.1, \max\{|f(x_k) - m_{k-1}(s^k)|/\|\nabla f(x_{k-1})\|\}, 10^{-3}\}\n
\eta_0 = 0.1.
\end{array}
\right.
\]
with \( m_{k-1}(s_k) \) given in (31). This choice is made according to (30). Finally, we choose the sample size \( D_k \) in SINA_FT_Dk as:

\[
D_k = \left[ \max\{c_0 D_0, \min\{c_1 \min\left\{ \frac{1}{\eta_k}, \frac{1}{\|\nabla f(x_k)\|_2^2} \right\}, N\} \right] \quad c_0, c_1 > 0
\]

The above rule is based on inequalities (36) and (38). The idea is to choose \( D_k \) inversely proportional to \( \gamma_k^2 \) as suggested by the bounds in (36). The choice of costants \( c_0 \) and \( c_1 \) depends on the convergence behaviour of CG method. In fact, if CG converges fast large values of \( D_k \) should be used, as the lost in the convergence rate due to less accurate second order information is not compensated by the reduced cost of Hessian-vector products. On the other hand, when CG is slower, smaller values of \( D_k \) must be used as a large number of matrix-vector products are needed. Then, \( c_0 = 1 \), \( c_1 = 0.05 \) in case at the previous iteration CG needed more than 20 iterations. Otherwise we set \( c_0 = 2 \) and \( c_1 = 1 \). Moreover, \( D_0 \) is set to \( 0.1N \). This choice is motivated by the fact that we allow \( D_k \) to change and increase, then we can start with a small \( D_k \) leaving the method free to adaptively modify it.

In all the subsampled methods the set \( D_k \) is chosen randomly using the Matlab function \texttt{randperm}. All the methods under comparison are in the framework of Algorithm GIN-R, i.e. the nonmonotone linesearch (8), with \( \epsilon = 10^{-4} \) and \( \nu_k = \max(1, f(x^0)/k^{1.1}) \) is applied.

The problem we consider is the binary classification problem. We suppose to have at disposal a training set composed of pairs \( \{ (a_i, b_i) \} \) with \( a_i \in \mathbb{R}^n \), \( b_i \in \{-1, +1\} \) and \( i = 1, \ldots , N \), where \( b_i \) denotes the correct sample classification. We perform a logistic regression, then we consider as a training objective function the logistic loss with \( \ell_2 \) regularization, i.e. in problem (1) we have

\[
f_i(x) = \log c(x, \xi_i) + \lambda \|x\|^2, \quad c(x, \xi_i) = 1 + e^{-b_i a_i^T x}
\]

where \( \xi_i = (a_i, b_i) \). Furthermore, the gradients and the Hessians have special forms,

\[
\nabla f_i(x) = \left( \frac{1 - c(x, \xi_i)}{c(x, \xi_i)} \right) b_i a_i + 2 \lambda x
\]

\[
\nabla^2 f_i(x) = -\left( \frac{1 - c(x, \xi_i)}{c^2(x, \xi_i)} \right) a_i a_i^T - 2 \lambda I.
\]

Note that the evaluation of the full function \( f \) requires the evaluation of the quantities \( c(x, \xi_i) - 1 \) and \( a_i \), for \( i = 1, \ldots , n \) and once these quantities have been computed they can be used for evaluating \( \nabla f_i(x) \) and \( \nabla^2 f_i(x), i = 1, \ldots , n \). Then, due to the form of gradient and Hessian of each \( f_i \) given in (56) and (57), the evaluation of \( \nabla f_i(x) \) comes from free and the evaluation of \( \nabla^2 f_i \) times a vector is as expensive as evaluating \( f_i(x) \). Then, we evaluate the performance of the methods under comparison in terms of full function evaluations (FEV). We underline that one \texttt{pcg} iteration costs as \( D_k \) FEV. Finally, in (55) we set \( \lambda = 1/N \).

We used the following three datasets...
- CINA0 [13], $N = 16033$, $n = 132$;
- Mushrooms [29], $N = 5000$, $n = 112$;
- Gisette [29], $N = 6000$, $n = 5000$.

We stopped the methods under comparison when

$$||\nabla f(x_k)|| \leq 10^{-4}$$

or when a maximum number of 50 nonlinear iterations is reached. In order to compute the training error we compute the minimizer $x^*$ with the full Newton method to a tight accuracy, i.e we run it until $||\nabla f(x_k)||$ is less than $10^{-8}$.

We begin by reporting our result with the mushrooms dataset. We first underline that the linear algebra phase for this test is not demanding and the average number of required CG iteration is small, as an example it is around 10 when the adaptive choice of the forcing term is used. In Figure 1 we plot $f_k - f(x^*)$ (training error) versus iterations (left) and versus function evaluations FEV (right). We can observe that as expected the full Inexact Newton (FIN) method is the fastest procedure. Moreover, the adaptive choice of the forcing terms seems to speed up the subsampled inexact procedure. Finally, remarkably, the procedure employing both the adaptive choice of $\eta_k$ and $D_k$ seems to work quite well. Indeed it is slower than SINA_FT in the first stage of the convergence history as it uses a smaller Hessian sample set. In the last stage of the procedure it becomes faster as the sample size increases (see Figure 2). On the other hand, if we look to the computational cost of the procedures we can observe that SINA_FT_Dk outperforms all the procedures under comparison. We also note that SINA_FT outperforms SIN. Overall these results show the efficiency of the proposed adaptive strategies.

To give more insight on our adaptive choices we plot in Figure 2 the values of the forcing terms $\eta_k$’s (left) and the value of $D_k$ (right) versus iterations using the SINA_FT_Dk. We can observe that whenever $\eta_k$ becomes smaller $D_k$ increases and
the approximation of the hessian improves. Moreover, we note that the adaptive pro-
dure allows the Hessian sample size to decrease when the model does not approximate sufficiently well the function and correspondingly the forcing term is increased.

We also compare, in Figure 3, our adaptive procedure SINA_FT_Dk with SIN_cg5. It is interesting to note that the adaptive procedure is faster than SIN_cg5 and greater accuracy can be reached. In terms of FEV SIN_cg5 is slightly better than SINA_FT_Dk till the accuracy of $10^{-4}$. If more accuracy is needed the adaptive procedure is clearly preferable.

Finally, in Figure 4 we compare the behaviour of testing errors versus iterations (left) and FEVs (right) along the sequences generated by full Newton method FIN, the adaptive procedure SINA_FT_Dk and SIN_cg5 method. We evaluated the testing error as follows. Let $x^k$ be the approximation computed by a method using the data in the training set. Then, $x^k$ is used to classify the samples in the testing set made up of $\bar{N} = 3124$ instances $z_i, i = 1, \ldots, \bar{N}$ and corresponding $b_i$'s. The clas-
Classification error at iteration $k$ is defined as $\frac{1}{N} \sum_{i=1}^{N} \log(1 + \exp(-b_i z^T x^k))$. We observe that the three methods reached almost the same testing errors; the subsampled approaches require a lower computational cost and again the adaptive procedure outperforms $\text{SINA}_{cg5}$. We also observe that the testing error quickly steadies. This enlightens that the stopping tolerance we used is tight enough and going on with the iterations would not decrease further the testing error.

Let us consider the CINA0 dataset. In Figure 5 we plot the training error versus iterations (left) and versus F-evaluations (right). The full Inexact Newton method $\text{FIN}$ is the fastest one, as expected. On the other hand, also in this case, it is the most expensive. The behaviour of the subsampled procedures is that desirable, with the adaptive procedures that outperforms the $\text{SINA}_{FT}$ in terms of FEV. Note that $\text{SINA}_{FT/Dk}$ is slower than the other two subsampled procedures, but less costly, as expected. In fact this is a problem where the linear algebra phase is more demanding, the average number of CG iteration using $\text{SINA}_{FT}$ is about 70, and small values of $D_k$ are used to limit the overall computational cost (see Figure 6, left). Since the convergence of CG is slow $\text{SINA}_{cg5}$ is not able to converge with a reasonable rate and it can only provide a very rough accuracy (see Figure 6, right).

We finally show the results obtained with the Gisette dataset, where $n$ is larger than in the previous tests. In Figure 7 we compare $\text{SINA}_{FT/Dk}$, $\text{SIN}$ and $\text{SINA}_{FT}$ in terms of function evaluations FEV (left) and also report the behaviour of the sample size along the iterations (right). We can observe that $\text{SIN}_{FT}$ and $\text{SINA}_{FT/Dk}$ are less expensive and more accurate than $\text{SIN}$, however within 50 nonlinear iterations $\text{SINA}_{FT/Dk}$ is not able to produce an approximation as accurate as that provided by $\text{SINA}_{FT}$. In fact, the convergence is slow due to the small size of the Hessian subsample set. However, note that if a training error of the order of $10^{-2}$ is enough, $\text{SINA}_{FT/Dk}$ is the method of choice as it is less expensive. In case a greater accuracy is needed $D_0 = 0.3N$ should be used in $\text{SINA}_{FT/Dk}$. Figure 8 refers to this choice of $D_0$. In this case $\text{SINA}_{FT/Dk}$ outperforms both $\text{SIN}$ and $\text{SINA}_{FT}$. We also report the values of $D_k$, $\eta_k$ and the number of cg-iterations at each Newton iteration. Plots show
Figure 5: CINA0 dataset: training error versus iteration (left) and versus FEV (right)

Figure 6: CINA0: $D_k$ values in $SINA_{FT\_DK}$ (left), training error (right)
Figure 7: Gisette dataset: training error versus FEV (left), $D_k$'s versus Newton iteration (right), $\eta_k$'s versus Newton iterations (bottom-left) and cg-iterations versus Newton iterations (bottom-right)

that we use small values of $D_k$ and large values of $\eta_k$ till the last stage of convergence. Since we are solving linear systems with a rough accuracy, the number of cg-iterations is reasonable, except for a few occurrences, considering that $n = 5000$ and we are not employing a preconditioner.

Acknowledgements
The authors are grateful to the anonymous referees for their suggestions, which lead to significant improvement of the manuscript.

References

[1] Bastin F., Trust-Region Algorithms for Nonlinear Stochastic Programming and Mixed Logit Models, PhD thesis, University of Namur, Belgium, 2004.

[2] Bastin F., Cirillo C., Toint P.L., An adaptive Monte Carlo algorithm for computing mixed logit estimators, Computational Management Science 3(1), 55-79, 2006.

[3] Bastin F., Cirillo C., Toint P.L., Convergence theory for nonconvex stochastic programming with an application to mixed logit, Mathematical Programming, 108, 207-234, 2006.

[4] Bellavia, S., Gurioli G., Morini B., Theoretical study of an adaptive cubic regularization method with dynamic inexact Hessian information, arXiv preprint arXiv:1808.06239, 2018.

[5] Berahas A. S., Bollapragada R., Nocedal J., An Investigation of Newton-Sketch and Subsampled Newton Methods, arXiv preprint arXiv:1705.06211, 2018.
Figure 8: Gisette dataset ($D_0 = 0.3N$ in SINA_FT_DK): training error versus FEV (top-left), $D_k$'s versus Newton iteration (top-right), $\eta_k$'s versus Newton iterations (bottom-left) and cg-iterations versus Newton iterations (bottom-right)
[6] Birgin, E.G., Krejić, N., Martínez, J.M., On the employment of Inexact Restoration for the minimization of functions whose evaluation is subject to programming errors, *Mathematics of Computation* (to appear).

[7] Bollapragada, R., Byrd, R., Nocedal, J., Exact and Inexact Subsampled Newton Methods for Optimization, *MA Journal Numerical Analysis*, 2018.

[8] Bottou, L., Curtis F.C., Nocedal, J. Optimization Methods for Large-Scale Machine Learning, *arXiv preprint arXiv:1606.04838v2 [stat.ML]*, 2017.

[9] Byrd R.H., Chin G.M., Nocedal J., Wu Y., Sample size selection in optimization methods for machine learning, *Mathematical Programming*, 134(1), 127-155, 2012.

[10] Byrd R.H., Chin G.M., Neveitt W., Nocedal J., On the Use of Stochastic Hessian Information in Optimization Methods for Machine Learning, *SIAM Journal on Optimization*, 21 (3), 977-995, 2011.

[11] Byrd R.H., Hansen S.L., Nocedal J., Singer Y., A Stochastic Quasi-Newton Method for Large-Scale Optimization, *SIAM Journal on Optimization*, 26(2), 1008-1021, 2016.

[12] Chernoff H., A measure of the asymptotic efficiency for tests of a hypothesis based on the sum of observations, *Annals of Mathematical Statistics* 23(4), 493-507, 1952.

[13] Causality workbench team, A marketing dataset, [http://www.causality.inf.ethz.ch/data/CINA.html](http://www.causality.inf.ethz.ch/data/CINA.html) 2008.

[14] Dembo, R.S. Eisenstat, S.C. Steinhaug, T., Inexact Newton method, *SIAM Journal on Numerical Analysis* 19(2), 400-409, 1982.

[15] Deng G., Ferris, M. C., Variable-Number Sample Path Optimization, *Mathematical Programming*, 117 (1-2), 81-109, 2009.

[16] Eisenstat, S.C., Walker, H.F., Choosing the Forcing Terms in an Inexact Newton Method, *SIAM Journal on Scientific Computing*, 17(1), 16-32, 1996.

[17] Fountoulakis, K., Godzio, J., A second order method for strongly convex $\ell_1$—regularization problems, *Mathematical Programming*, 156, 189-219, 2016.

[18] Erdogdu M. A., Montanari A., Convergence rates of sub-sampled Newton methods, *NIPS’15 Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2, 3052-3060, 2015.

[19] Friedlander M.P., Schmidt M., Hybrid deterministic-stochastic methods for data fitting, *SIAM Journal on Scientific Computing*, 34(3), 1380-1405, 2012.

[20] Grapiglia G.N., Sachs E.W., On the worst-case evaluation complexity of non-monotone line search algorithms, *Computational Optimization and Applications*, 68(3), 555-577, 2017.
[21] Hoeffding, W., Probability inequalities for sums of bounded random variables, *American Statistical Association Journal*, 58(103), 13-30, 1963.

[22] Jakovetić D., Xavier J., Moura J.M.F. Fast Distributed Gradient Methods, *IEEE Trans. Autom. Contr.*, 59(5), 1131-1146, 2014.

[23] Kelley, C.T., Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.

[24] Krejić N., Krklec N., Line search methods with variable sample size for unconstrained optimization, *Journal of Computational and Applied Mathematics* 245, 213-231, 2013.

[25] Krejić N., Krklec Jerinkić N., Nonmonotone line search methods with variable sample size, *Numerical Algorithms* 68(4), 711-739, 2015.

[26] Krejić, N., Martínez, J.M., Inexact Restoration approach for minimization with inexact evaluation of the objective function, *Mathematics of Computation*, 85 (2016) 1775-1791.

[27] Lee, Jason D., Yuekai Sun, Michael A. Sandres, Proximal Newton type methods for minimizing composite functions, *SIAM Journal on Optimization* 24, 3 (2014), 1420-1443.

[28] Li D.H., Fukushima M., A derivative-free line search and global convergence of Broyden-like method for nonlinear equations, *Optimization Methods and Software* 13, 181-201, 2000.

[29] Lichman M., UCI machine learning repository, https://archive.ics.uci.edu/ml/index.php, 2013.

[30] Minsker S., On some extensions of Bernstein’s inequality for self-adjoint operators, arXiv:1112.5448 [math.PR], 2017.

[31] Nash S.G., A survey of truncated-Newton methods, *Journal of Computational and Applied Mathematics* 124(1-2), 45-59, 2000.

[32] Nesterov, Y., Introductory Lectures on Convex Optimization: A Basic Course, Springer Science and Media, Vol. 87, 2013.

[33] Pasupathy R., On Choosing Parameters in Retrospective-Approximation Algorithms for Stochastic Root Finding and Simulation Optimization, *Operations Research* 58(4), 889-901, 2010.

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

[35] Polak E., Royset J.O., Efficient sample sizes in stochastic nonlinear programing, *Journal of Computational and Applied Mathematics* 217(2), 301-310, 2008.
[36] Roosta-Khorasani, F., Mahoney, M.W., Sub-Sampled Newton Methods I: Globally Convergent Algorithms, arXiv preprint, arXiv:1601.04737v3 [math.OC], 2016.

[37] Roosta-Khorasani, F., Mahoney, M.W., Sub-Sampled Newton Methods II: Local Convergence Rates, arXiv preprint, arXiv:1601.04738v3 [math.OC], 2016.

[38] Shapiro, D. Dentcheva, A. Ruszczynski, Lectures on stochastic programming: Modeling and theory, MPS/SIAM Series on Optimization 9, 2009.

[39] Tropp, J.A., User-Friendly Tail Bounds for Sums of Random Matrices, Foundations of Computational Mathematics, 12 (2012), 389-434.

[40] Xu P., Yang J., Roosta-Khorasani F., Räli C., and Mahoney M.W., Sub-sampled Newton methods with non-uniform sampling, Advances in Neural Information Processing Systems 30 (NIPS), 2530-2538, 2016.

[41] Xu P., Roosta-Khorasani F., and Mahoney M.W., Newton-Type Methods for Non-Convex Optimization Under Inexact Hessian Information, arXiv preprint, arXiv:1708.07164v3 [math.OC], 2018.

[42] Zhang H., Hager W., A nonmonotone line search technique and its application to unconstrained optimization SIAM Journal on Optimization, 4, 1043-1056, 2004.

5 Appendix

Proof of Lemma 2.6. Using the Mean Value Theorem we get the following inequality for the full Hessian.

\[ \| \nabla f_N(x^k + s^k) - \nabla f_N(x^k) - \nabla^2 f_N(x^k)s^k \| \leq \frac{L}{2} \| s^k \|^2. \]

Now,

\[ \| \nabla f_N(x^k + s^k) - \nabla f_N(x^k) - \nabla^2 f_D(x^k)s^k \| \leq \| \nabla f_N(x^k + s^k) - \nabla f_N(x^k) - \nabla^2 f_N(x^k)s^k \| 
+ \| \nabla^2 f_D(x^k) - \nabla^2 f_N(x^k) \| \| s^k \| \]

\[ \leq \| s^k \| (h(D_k, x^k) + \frac{1}{2}L\| s^k \|). \]

Then,

\[ \| \nabla f_N(x^k + s^k) \| \leq \| \nabla f_N(x^k) + \nabla^2 f_D(x^k)s^k \| + \| \nabla f_N(x^k + s^k) - \nabla f_N(x^k) - \nabla^2 f_D(x^k)s^k \| 
\leq \eta \| \nabla f_N(x^k) \| + \| s^k \| (\frac{1}{2}L\| s^k \| + h(D_k, x^k)) \]

and, by Lemma 2.2

\[ \| \nabla f_N(x^k + s^k) \| \leq \eta \| \nabla f_N(x^k) \| + \frac{1}{\lambda_1} \| \nabla f_N(x^k) \| (\frac{1}{2\lambda_1}L\| \nabla f_N(x^k) \| + h(D_k, x^k)) \]
Proof of Lemma 2.7. By Lemma 2.2 and (5) we have
\[ \|x^k + s^k - x^*\| \leq \|x^k - x^*\| + \|s^k\| \leq \|x^k - x^*\| + \lambda_1^{-1} \|\nabla f_N(x^k)\| \]
\[ \leq \|x^k - x^*\| + \lambda_1^{-1} \lambda_n \|x^k - x^*\| \leq (1 + \lambda_1^{-1} \lambda_n) \|x^k - x^*\| \leq \delta^*. \]

Proof of Lemma 3.1. Let us define \( Y_i(x) = \frac{(H_i(x) - \nabla^2 f_N(x))}{D} \) where \( H_i(x) \) is a randomly chosen Hessian. Then,
\[ \sum_{i=1}^{D} Y_i(x) = \nabla^2 f_D(x) - \nabla^2 f_N(x). \]
Also, notice that \( E(Y_i(x)) = 0 \) and that the Weyl’s inequality yields \( \lambda_{\text{max}}(Y_i(x)) \leq \lambda_n/D, \lambda_{\text{min}}(Y_i(x)) \geq -\lambda_n/D. \) Then, \( \|Y_i(x)\| \leq \lambda_n/D \) and
\[ \bar{\sigma}^2(x) := \sum_{i=1}^{D} E(Y_i^2(x)) \leq \sum_{i=1}^{D} E(\|Y_i(x)\|^2) \leq \frac{\lambda_n^2}{D}. \]
Then, using the Bernstein’s inequality (see [39, Theorem 1.6]) we derive
\[ P(\|\nabla^2 f_D(x) - \nabla^2 f_N(x)\| \geq \gamma) \leq 2n \exp \left( -\frac{\gamma^2/2}{\lambda_n^2/D + (\lambda_n/D)\gamma/3} \right). \]
This yields the bound. ☐