Adaptive Newton Sketch: Linear-time Optimization with Quadratic Convergence and Effective Hessian Dimensionality

Jonathan Lacotte [1] **Yifei Wang** [1] **Mert Pilanci** [1]

Abstract

We propose a randomized algorithm with quadratic convergence rate for convex optimization problems with a self-concordant, composite, strongly convex objective function. Our method is based on performing an approximate Newton step using a random projection of the Hessian. Our first contribution is to show that, at each iteration, the embedding dimension (or sketch size) can be as small as the effective dimension of the Hessian matrix. Leveraging this novel fundamental result, we design an algorithm with a sketch size proportional to the effective dimension and which exhibits a quadratic rate of convergence. This result dramatically improves on the classical linear-quadratic convergence rates of state-of-the-art sub-sampled Newton methods. However, in most practical cases, the effective dimension is not known beforehand, and this raises the question of how to pick a sketch size as small as the effective dimension while preserving a quadratic convergence rate. Our second and main contribution is thus to propose an adaptive sketch size algorithm with quadratic convergence rate and which does not require prior knowledge or estimation of the effective dimension: at each iteration, it starts with a small sketch size, and increases it until quadratic progress is achieved. Importantly, we show that the embedding dimension remains proportional to the effective dimension throughout the entire path and that our method achieves state-of-the-art computational complexity for solving convex optimization programs with a strongly convex component. We discuss and illustrate applications to linear and quadratic programming, as well as logistic regression and other generalized linear models.

[1]Department of Electrical Engineering, Stanford University. Correspondence to: Jonathan Lacotte <lacotte@stanford.edu>, Yifei Wang <wangyf18@stanford.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

1. Introduction

We consider a composite optimization problem of the form

$$x^* := \operatorname*{argmin}_{x\in\mathbb{R}^d} \{f(x) := f_0(x) + g(x)\} , \qquad (1)$$

where $f_0, g : \mathbb{R}^d \to \overline{\mathbb{R}}$ are both closed, twice differentiable convex functions. Here, we denote $\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$ and by $\mathbf{dom}\, f$ the domain of f. We are interested in the structured setting where forming the Hessian matrix $\nabla^2 f_0(x)$ is prohibitively expensive, but we have available at small computational cost a Hessian matrix square-root $\nabla^2 f_0(x)^{1/2}$, that is, a matrix $\nabla^2 f_0(x)^{1/2}$ of dimensions $n \times d$ such that $(\nabla^2 f_0(x)^{1/2})^\top \nabla^2 f_0(x)^{1/2} = \nabla^2 f_0(x)$ for some integer $n \geqslant d$, and n eventually very large. Moreover, we assume the function g to be μ-strongly convex, i.e., $\nabla^2 g(x) \succeq \mu I_d$.

Large-scale optimization problems of this form are becoming ever more common in applications, due to the increasing dimensionality of data (e.g., genomics, medicine, high-dimensional models). Typically, the function f_0 may represent an objective value we aim to minimize over a convex set $\mathcal{C} \subseteq \mathbb{R}^d$, that is, we aim to solve $\min_{x\in\mathcal{C}} f_0(x)$. A common practice to turn this constrained optimization problem into an unconstrained one is to add to the objective function a *penalty* or barrier function $g(x)$ which encodes $\mathcal{C}$ (e.g., logarithmic barrier functions for polyhedral constraints or ℓ_p-norm regularization for ℓ_p-ball constraints). In many cases of practical interest, a matrix square-root $\nabla^2 f_0(x)^{1/2}$ can be computed efficiently. For instance, in the broad context of empirical risk minimization, the function f_0 has the separable form $f_0(x) = \sum_{i=1}^m \ell_i(a_i^\top x)$ where the functions ℓ_i are twice-differentiable and convex. In this case, a suitable Hessian matrix square root is given by the $n \times d$ matrix $\nabla^2 f_0(x)^{1/2} := \mathbf{diag}(\ell_i''(a_i^\top x)^{1/2})\, A$. On the other hand, we assume that the Hessian of the function g is well-structured, so that its computation is relatively cheap in comparison to that of f_0. For instance, if the constraint set is the unit simplex (i.e., $x \geqslant 0$ and $\mathbf{1}^\top x \leqslant 1$), then the Hessian of the associated logarithmic barrier function is a diagonal matrix plus a rank one matrix. Other examples include problems for which g has a separable structure such as typical regularizers for ill-posed inverse problems (e.g., graph regularization $g(x) = \frac{1}{2}\sum_{i,j\in E}(x_i - x_j)^2$, ℓ_p-norms with $p > 1$ or approximations of ℓ_1-norm).

Second-order methods such as the Newton's method enjoy superior convergence in both theory and practice compared to first-order methods, that is, quadratic convergence rate versus $1/T^2$ for accelerated gradient descent. A common issue in first-order methods is the tuning of step size (Asi & Duchi, 2019), whose optimal choice depends on the strong convexity and smoothness of the underlying problem. In contrast, whenever the objective function f is *self-concordant*, then Newton's method has the appealing property of being invariant to rescaling and coordinate transformations, is independent of problem-dependent parameters, and thus needs little or no tuning of algorithmic hyperparameters. More precisely, we recall that, given a current iterate x, the standard Newton's method computes the Hessian matrix $H(x)$ and the descent direction $v_{\rm ne}$ defined as

$$H(x) := \nabla^2 f_0(x) + \nabla^2 g(x)\,, \tag{2}$$
$$v_{\rm ne} := -H(x)^{-1}\nabla f(x)\,. \tag{3}$$

Given a step size $s > 0$, it then uses the update

$$x_{\rm ne} := x + s\, v_{\rm ne}\,. \tag{4}$$

Despite these advantages, Newton's method requires, at each iteration, forming and solving the high-dimensional linear system $H(x)v_{\rm ne} = -\nabla f(x)$, which has complexity scaling as $\mathcal{O}(nd^2)$, and this becomes prohibitive in large-scale settings. To address this numerical challenge, a multitude of different approximations to Newton's method have been proposed in the literature. Quasi-Newton methods (e.g, DFP, BFGS and their limited memory versions (Nocedal & Wright, 2006)) are computationally cheaper, but their convergence guarantees require stronger assumptions and are typically much weaker than those of Newtons's method. On the other hand, random projections are an effective way of performing dimensionality reduction (Vempala, 2005; Mahoney, 2011; Drineas & Mahoney, 2016), and many random projection (or *sketching*) based algorithms were designed to reduce the cost of solving the linear Newton system. For instance, the respective methods in (Gower et al., 2019) and (Lacotte et al., 2019) embed the optimization variable into a lower dimensional subspace, so that solving the Newton system becomes cheaper; (Qu et al., 2016) propose to solve an approximate Newton system based on random principal sub-matrices of a global upper bound on the Hessian; (Doikov & Richtárik, 2018) address a common setting, that of block-separable convex optimization problems, and propose a method combining the ideas of randomized coordinate descent with cubic regularization (Nesterov, 2012; Nesterov & Polyak, 2006).

Our work builds specifically on a generic method, that is, the Newton sketch (Pilanci & Wainwright, 2017), which is based on a structured random embedding of the Hessian matrix $H(x)$. Formally, given a sketch size m such that $m \ll n$ and an embedding matrix $S \in \mathbb{R}^{m\times n}$ to be precised, the Newton sketch computes the approximate Hessian $H_S(x)$ and the approximate descent direction $v_{\rm nsk}$ defined as

$$H_S(x) := (\nabla^2 f_0(x)^{\frac{1}{2}})^\top S^\top S \nabla^2 f_0(x)^{\frac{1}{2}} + \nabla^2 g(x)\,, \tag{5}$$
$$v_{\rm nsk} := -H_S(x)^{-1}\nabla f(x)\,. \tag{6}$$

Given a step size $s > 0$, it then uses the update

$$x_{\rm nsk} := x + s\, v_{\rm nsk}\,. \tag{7}$$

For classical embeddings (e.g., sub-Gaussian, randomized orthogonal systems), it has been shown by (Pilanci & Wainwright, 2017) that, in general, a sketch size $m \asymp d$ is sufficient for the Newton sketch to achieve a linear-quadratic convergence rate with high probability (w.h.p.).

Contributions. Our first key contribution is to show that, under the assumption that g is μ-strongly convex, the scaling $m \asymp \overline{d}_\mu \log(\overline{d}_\mu)/\delta$ is sufficient for the Newton sketch to achieve a δ-accurate solution at a *quadratic* convergence rate with high probability. More generally, we show that convergence is geometric provided that m scales appropriately in terms of $\overline{d}_\mu$. Here, the critical quantity $\overline{d}_\mu$ is the effective (Hessian) dimension, defined as

$$\overline{d}_\mu := \sup_{x\in\mathcal{S}(x_0)} d_\mu(x)\,, \tag{8}$$

where x_0 is the initial point of our algorithm, $\mathcal{S}(x_0)$ is the sublevel set of f at x_0, and

$$d_\mu(x) := \operatorname{trace}(\nabla^2 f_0(x)(\nabla^2 f_0(x) + \mu I_d)^{-1}) \tag{9}$$

is the *local* effective dimension. Importantly, it always holds that $d_\mu(x) \leqslant \overline{d}_\mu \leqslant \min\{n, d\} = d$. In many applications, the effective dimension is substantially smaller than the ambient dimension d (Bach, 2013; Alaoui & Mahoney, 2015; Yang et al., 2017). However, in order to pick m in terms of $\overline{d}_\mu$ which is usually unknown and then achieve computational and memory space savings, it is necessary to estimate $\overline{d}_\mu$. There exist randomized techniques for precise estimation of $d_\mu(x)$, but they provably work under stringent assumptions, e.g., $d_\mu(x)$ very small (e.g., see Theorem 60 in (Avron et al., 2017)). In the context of ridge regression, (Lacotte & Pilanci, 2020) proposed a sketching-based method with adaptive (time-varying) sketch size scaling as the effective dimension, and without prior knowledge or estimation of it. Starting with a small sketch size, it checks at each iteration whether enough progress is achieved by the update. If not, it doubles the sketch size. The time and memory complexities of this method to return a *certified* δ-accurate solution w.h.p. scale in terms of the effective dimension, i.e., it takes time $\mathcal{O}(nd\log^2(\overline{d}_\mu)\log(d/\delta))$ with a sketch size $m \lesssim \overline{d}_\mu \log(\overline{d}_\mu)$ for large values of n. This

significantly improves on usual standard randomized preconditioning methods (Rokhlin & Tygert, 2008; Avron et al., 2010; Meng et al., 2014) which require $m \gtrsim d$.

In a vein similar to this adaptive ridge regression solver, our second key contribution is to propose an adaptive sketch size version of the effective dimension Newton sketch. Importantly, we prove that the adaptive sketch size scales in terms of $\overline{d}_\mu$. Furthermore, our adaptive method offers the possibility to the user to choose the convergence rate, from linear to quadratic.

Other related works. Recent studies in the literature on randomized second-order and Sub-sampled Newton methods (Byrd et al., 2011; Bollapragada et al., 2019; Roosta-Khorasani & Mahoney, 2019; Berahas et al., 2020) show that picking an embedding dimension proportional to d and possibly smaller than d under certain conditions do work empirically in many settings (Xu et al., 2016; 2020; Wang et al., 2018). The recent work by (Li et al., 2020) provides a more precise understanding of these phenomena. In the context of empirical risk minimization with ℓ_2^2-regularization, they show that the subsampled Newton method with $m \asymp \overline{d}_\mu$ data points is enough to guarantee convergence. However, differently from our work, their method needs to estimate the effective dimension at each iteration. Furthermore, their convergence guarantees severely depend on the condition number of the problem (e.g., see their Theorems 1 and 2), whereas our results are independent of condition numbers and only involve the relevant dimensions of the problem $(n, d, \overline{d}_\mu)$ and the target accuracy. Besides effective dimension based sampling, sketching-based methods are used in the context of distributed optimization where due to stringent memory and/or communication constraints, reducing the number of iterations and/or the size of second-order information is critical (Shamir et al., 2014; Derezinski et al., 2020; Bartan & Pilanci, 2020).

1.1. Notations and background

A closed convex function $\varphi : \mathbb{R}^d \to \overline{\mathbb{R}}$ is *self-concordant* if $|\varphi'''(x)| \leqslant 2\,(\varphi''(x))^{3/2}$. This definition extends to a closed convex function $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ by imposing this requirement on the univariate functions $\varphi_{x,y}(t) := f(x + ty)$ for all choices of x, y in the domain of f. Self-concordance is a typical assumption for the analysis of the classical Newton's method, in order to obtain convergence results which are independent of unknown problem parameters (e.g., strong convexity, smoothness or Lipschitz constants; see the books by (Nesterov, 2003) or (Boyd & Vandenberghe, 2004) for further background), and this encompasses many widely used functions in practice, e.g., linear, quadratic, negative logarithm. Hence, in this work, we assume that *f_0 and g are self-concordant functions.*

The choice of the sketching matrix $S \in \mathbb{R}^{m\times n}$ is critical for statistical and computational performances. The well-structured subsampled randomized Hadamard transform (SRHT) (Ailon & Chazelle, 2006) usually serves as a reference for comparing sketching algorithms thanks to its strong subspace embedding properties (Mahoney, 2011; Drineas & Mahoney, 2016; Dobriban & Liu, 2019; Lacotte et al., 2020) and fast sketching time $\mathcal{O}(nd \log m)$ compared to the classical sketching cost $\mathcal{O}(ndm)$ of sub-Gaussian embeddings. Another typical choice is the sparse Johnson-Lindenstrauss transform (SJLT) (Nelson & Nguyên, 2013; Woodruff et al., 2014) with, for instance, one non-zero entry per column. With $A \in \mathbb{R}^{n\times d}$, a sketch SA is then much faster to compute (it takes time $\mathcal{O}(\mathbf{nnz}(A))$) at the expense of weaker subspace embedding properties.

1.2. Organization of the paper

In Section 2, we introduce critical quantities and preliminary results for both the implementation of our algorithms and their analysis. We show that the approximate Newton direction $v_{\rm nsk}$ is close to the exact one $v_{\rm ne}$, provided that the sketch size scales in terms of $\overline{d}_\mu$. In Section 3, we formally introduce our (non-adaptive) effective dimension Newton sketch algorithm (see Algorithm 1), and we present several relevant applications. Assuming knowledge of $\overline{d}_\mu$, we prove that its convergence rate is geometric. In Section 4, we introduce an adaptive version of Algorithm 1 (see Algorithm 2): importantly, it does not require knowledge of $\overline{d}_\mu$, but still guarantees geometric convergence as well as low memory complexity in terms of $\overline{d}_\mu$. We summarize our complexity guarantees in Table 1 and compare to standard first- and second-order methods and to the original Newton sketch algorithm (Pilanci & Wainwright, 2017) whose implementation and guarantees are agnostic to the effective dimension of the problem. Finally, we show in Section 5 the empirical benefits of our adaptive method, compared to several standard optimization baselines.

2. Preliminaries

Critical to our algorithms and their analysis are the Newton and approximate Newton decrements, defined as

$$\lambda_f(x) := \left(\nabla f(x)^\top H(x)^{-1} \nabla f(x)\right)^{\frac{1}{2}}, \tag{10}$$

$$\widetilde{\lambda}_f(x) := \left(\nabla f(x)^\top H_S(x)^{-1} \nabla f(x)\right)^{\frac{1}{2}}. \tag{11}$$

Importantly, for a self-concordant function f, the optimality gap at any point $x \in \mathbf{dom}\, f$ is bounded in terms of the Newton decrement as

$$f(x) - f(x^*) \leqslant \lambda_f(x)^2\,. \tag{12}$$

Due to the expensive cost of computing the Newton decrement $\lambda_f(x)$ as opposed to $\widetilde{\lambda}_f(x)$, we will aim to charac-

terize, w.h.p. over the randomness of the sketching matrix, similar optimality bounds and properties with $\widetilde{\lambda}_f(x)$.

Given $x \in \mathbf{dom}\, f$, a sketch size $m \geqslant 1$, a random embedding $S \in \mathbb{R}^{m\times d}$ and a sampling precision parameter $\varepsilon > 0$, we consider the following probability event which is critical to our convergence guarantees,

$$\boxed{\mathcal{E}_{x,m,\varepsilon} := \Big\{(1-\frac{\varepsilon}{2})I_d \preceq C_S \preceq (1+\frac{\varepsilon}{2})I_d\Big\}}, \quad (13)$$

where $C_S := H^{-\frac{1}{2}} H_S H^{-\frac{1}{2}}$, $H \equiv H(x)$ and $H_S \equiv H_S(x)$. In words, when $\mathcal{E}_{x,m,\varepsilon}$ holds true, the matrix $H^{-1/2}H_S H^{-1/2}$ is a close approximation of the identity, i.e., $H^{-1/2}H_S H^{-1/2} \approx H^{-1/2}HH^{-1/2} = I_d$. The next result bounds the probability for this event to hold for different choices of the sketching matrix.

Lemma 1. *Let $\varepsilon \in (0, 1/4)$ and $p \in (0, 1/2)$. It holds that $\mathbb{P}(\mathcal{E}_{x,\varepsilon,m}) \geqslant 1 - p$, provided that $m = \Omega(d_\mu(x)^2/(\varepsilon^2 p))$ for the SJLT with single nonzero element in each column, and, $m = \Omega((d_\mu(x) + \log(1/\varepsilon p)\log(d_\mu(x)/p))/\varepsilon^2)$ for the SRHT.*

We show next that conditional on $\mathcal{E}_{x,m,\varepsilon}$, the approximate Newton decrement $\widetilde{\lambda}_f(x)$ is close to $\lambda_f(x)$, as well as the approximate Newton direction v_{nsk} to the exact one v_{ne}.

Theorem 1 (Closeness of Newton decrements)**.** *Let $\varepsilon \in (0, 1/4)$. Conditional on the event $\mathcal{E}_{x,m,\varepsilon}$, it holds that*

$$\|v_{\mathrm{ne}} - v_{\mathrm{nsk}}\|_{H(x)} \leqslant \varepsilon\, \|v_{\mathrm{ne}}\|_{H(x)}\,, \quad (14)$$

$$\sqrt{1-\varepsilon}\,\lambda_f(x) \leqslant \widetilde{\lambda}_f(x) \leqslant \sqrt{1+\varepsilon}\,\lambda_f(x)\,. \quad (15)$$

Given $\varepsilon \in (0, 1/4)$, we introduce positive parameters a, b such that $1 - \frac{1}{2}\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2 \geqslant a$, which we use for backtracking line-search (see Algorithm 1 for details). Furthermore, we define the parameters

$$\eta := \frac{1}{8}\left(1 - \frac{1}{2}\Big(\frac{1+\varepsilon}{1-\varepsilon}\Big)^2 - a\right)\Big/\Big(\frac{1+\varepsilon}{1-\varepsilon}\Big)^3,$$
$$\nu := ab\,\frac{\eta^2}{1 + \frac{1+\varepsilon}{1-\varepsilon}\,\eta}\,.$$

The next results aim to describe the empirical behavior of our methods. As for the classical Newton's method, we distinguish two phases. The algorithm follows a first phase with constant additive decrease in objective value. In a second phase, it converges faster, i.e., the Newton decrement converges to zero at a geometric rate up to quadratic for an appropriate choice of the hyperparameters.

Lemma 2 (First phase decrement)**.** *Let $\varepsilon \in (0, 1/4)$. Suppose that $\mathcal{E}_{x,m,\varepsilon}$ holds true and that $\widetilde{\lambda}_f(x) > \eta$. Then, we have that*

$$f(x_{\mathrm{nsk}}) - f(x) \leqslant -\nu\,. \quad (16)$$

We introduce the following numerical function which will prove to be useful to characterize the rate of convergence of our algorithms,

$$\alpha(\tau) := 0.57 + \frac{16^\tau}{15}\,. \quad (17)$$

It is easy to verify that $\alpha(\tau)^{1/\tau} \leqslant 2$ for $\tau \in (0, 1]$ and $\alpha(0) \leqslant \frac{16}{25}$.

Lemma 3 (Second phase decrement)**.** *Let $x \in \mathbf{dom}\, f$, $\tau \in [0, 1]$ and $\varepsilon \in (0, 1/4)$. Set $\varepsilon' = \varepsilon \min\{1, \lambda_f(x)^\tau\}$. We assume that the event $\mathcal{E}_{x,m,\varepsilon'}$ holds and that $\widetilde{\lambda}_f(x) \leqslant \eta$. Then, we have*

$$\lambda_f(x_{\mathrm{nsk}}) \leqslant \alpha(\tau)\,\lambda_f(x)^{1+\tau}\,. \quad (18)$$

Consequently, the progress is geometric for any $\tau \in (0, 1]$, i.e.,

$$\alpha(\tau)^{1/\tau}\,\lambda_f(x_{\mathrm{nsk}}) \leqslant \Big(\alpha(\tau)^{1/\tau}\,\lambda_f(x)\Big)^{1+\tau}\,. \quad (19)$$

On the other hand, the progress is linear for $\tau = 0$, i.e.,

$$\lambda_f(x_{\mathrm{nsk}}) \leqslant \frac{16}{25}\,\lambda_f(x)\,. \quad (20)$$

We conclude this section with a simple technical lemma which characterizes a sufficient number of iterations before termination, under geometric convergence.

Lemma 4 (Geometric convergence and sufficient iteration number)**.** *Let $\delta \in (0, 1)$, $\alpha > 0$, $\tau \in (0, 1]$, and $\{\beta_t\}_{t\geqslant 0}$ be a sequence of positive numbers such that $\beta_0 \leqslant \eta$, $\eta\alpha^{1/\tau} < 1$, $\sqrt{\delta}\alpha^{1/\tau} < 1$ and $\alpha^{1/\tau}\beta_{t+1} \leqslant (\alpha^{1/\tau}\beta_t)^{1+\tau}$ for all $t \geqslant 0$. Then, it holds that $\beta_t \leqslant \sqrt{\delta}$ for any $t \geqslant T_{\tau,\alpha,\delta}$ where*

$$T_{\tau,\alpha,\delta} := \Big\lceil \frac{1}{\log(1+\tau)}\,\log\left(\frac{1 + \frac{\tau\log(1/\delta)}{2\log(1/\alpha)}}{1 + \frac{\tau\log(1/\eta)}{\log(1/\alpha)}}\right)\Big\rceil\,. \quad (21)$$

Throughout this work, we will use the shorthand

$$\boxed{T_{\tau,\delta} \equiv T_{\tau,\alpha(\tau),\delta}}\,. \quad (22)$$

Note in particular that $T_{\tau,\delta} = \mathcal{O}(\log(\tau\log(1/\delta)))$ for small δ. Further, it holds that $\lim_{\tau\to 0} T_{\tau,\delta} \leqslant \lceil \frac{\log(1/\delta)}{\log(25/16)} \rceil$, which corresponds to the classical complexity of linear convergence with rate $16/25$.

3. Effective dimension Newton sketch

We formally introduce our effective dimension Newton sketch method in Algorithm 1. Algorithm 1 takes as inputs the phase 1 and phase 2 sketch sizes $\overline{m}_1$ and $\overline{m}_2$. As we will see in Theorem 2, sufficient values for $\overline{m}_1$ and $\overline{m}_2$

Algorithm 1 Effective dimension Newton sketch

Require: Initial point $x_0 \in \mathbf{dom} f$, threshold sketch sizes $\overline{m}_1$ and $\overline{m}_2$, initial sketch size $m_0 = \overline{m}_1$, line-search parameters (a, b), target accuracy $\delta > 0$, convergence rate parameter $\tau \in [0, 1]$ and sampling precision parameter $\varepsilon = 1/8$.

1: **for** $t = 0, \ldots$ **do**
2: Sample an $m_t \times n$ embedding S_t independent of $\{S_j\}_{j=0}^{t-1}$. Compute $v_{\rm nsk}$ and $\widetilde{\lambda}_f(x_t)$ based on S_t.
3: **if** $\widetilde{\lambda}_f(x_t)^2 \leqslant \frac{3}{4}\delta$ **then return** x_t.
4: Starting at $s = 1$: **while** $f(x_t + s v_{\rm nsk}) > f(x_t) + as\nabla f(x_t)^\top v_{\rm nsk}, \quad s \leftarrow bs$.
5: Update $x_{t+1} \leftarrow x_t + s\, v_{\rm nsk}$.
6: If $\widetilde{\lambda}_f(x_t) > \eta$, set $m_{t+1} = \overline{m}_1$. Otherwise, set $m_{t+1} = \overline{m}_2$.
7: **end for**

to guarantee convergence both depend on the effective dimension $\overline{d}_\mu$. Here and only for Algorithm 1, we make the idealized assumption that the quantity $\overline{d}_\mu$ is known. In contrast, we introduce in Section 4 an adaptive method that does not require knowledge of $\overline{d}_\mu$.

Theorem 2 (Geometric convergence guarantees of the Newton sketch)**.** *Let $\tau \in [0, 1]$, $\delta \in (0, 1/2)$ and $p_0 \in (0, 1/2)$. Set $\varepsilon = 1/8$. Then, the total number of iterations T_f and the total time complexity $\mathcal{C}$ for obtaining a δ-approximate solution $\widetilde{x}$ in function value (i.e., $f(\widetilde{x}) - f(x^*) \leqslant \delta$) via Algorithm 1 satisfy*

$$T_f \leqslant \overline{T} := \frac{f(x_0) - f(x^*)}{\nu} + T_{\tau, \frac{3}{8}\delta} + 1\,, \tag{23}$$

$$\mathcal{C} = \mathcal{O}\big(\overline{m}_2^2 d + nd \log \overline{m}_2\big)\, \overline{T}\,, \tag{24}$$

with probability at least $1 - p_0$, provided that $\overline{m}_1 \gtrsim \overline{d}_\mu + \log(\frac{\overline{T}}{p_0}) \log(\frac{\overline{d}_\mu \overline{T}}{p_0})$ and $\overline{m}_2 \gtrsim \delta^{-\tau}\left(\overline{d}_\mu + \log(\frac{\overline{T}}{p_0 \delta^{\tau/2}}) \log(\frac{\overline{d}_\mu \overline{T}}{p_0})\right)$ for the SRHT, whereas for the SJLT, it is sufficient to have $\overline{m}_1 \gtrsim \frac{\overline{d}_\mu^2 \overline{T}}{p_0}$ and $\overline{m}_2 \gtrsim \frac{\overline{d}_\mu^2 \overline{T}}{\delta^\tau p_0}$.

We draw some immediate consequences of Theorem 2, which will be useful for further discussions and comparisons of our complexity guarantees in Section 4.1. With the SRHT, consider the quadratic convergence case, i.e., $\tau = 1$. We pick a failure probability $p_0 \asymp \frac{1}{\overline{d}_\mu}$, and sketch sizes $\overline{m}_1 \asymp \overline{d}_\mu$ and $\overline{m}_2 \asymp \frac{\overline{d}_\mu \log(\overline{d}_\mu/\delta)}{\delta}$. We observe quadratic convergence with $T_f = \mathcal{O}(\log\log(\frac{1}{\delta}))$ iterations. Further, assuming that the sample size n is large enough for the sketching cost $\mathcal{O}(nd \log \overline{m})$ to dominate the cost $\mathcal{O}(\overline{m}^2 d)$ of solving the randomized Newton system, i.e., $n \gtrsim \frac{\overline{d}_\mu^2 \log(\overline{d}_\mu/\delta)}{\delta^2}$, then the total complexity results in

$$\mathcal{C} = \mathcal{O}\big(nd \log\big(\frac{\overline{d}_\mu}{\delta}\big) \log\log(\frac{1}{\delta})\big). \tag{25}$$

Similarly, we consider the linear convergence case, i.e., $\tau = 0$. For simplicity, suppose that $\overline{d}_\mu \gtrsim \log\log(1/\delta)$. We pick $p_0 \asymp \frac{1}{\overline{d}_\mu}$, and sketch sizes $\overline{m}_1 \asymp \overline{m}_2 \asymp \overline{d}_\mu$. We observe linear convergence with $T_f = \mathcal{O}(\log \frac{1}{\delta})$ iterations. Assuming again that the sample size n is large enough for the sketching cost to dominate the cost of solving the randomized Newton system, i.e., $n \gtrsim \overline{d}_\mu^2 / \log(\overline{d}_\mu)$, we obtain the total time complexity

$$\mathcal{C} = \mathcal{O}\left(nd \log(\overline{d}_\mu) \log(\frac{1}{\delta})\right). \tag{26}$$

We proceed with a similar discussion for the SJLT at the end of the proof of Theorem 2 deferred to the Appendix.

3.1. Some applications of the effective dimension Newton sketch

We discuss various concrete instantiations of the optimization problem (1) where the function g satisfies μ-strong convexity and for which forming the partially sketched Hessian $H_S(x)$ is amenable to fast computation.

Example 1 (Ridge regression)**.** *We consider the optimization problem*

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{2}\|Ax - b\|_2^2 + \frac{\mu}{2}\|x\|_2^2 \right\} \tag{27}$$

where $A \in \mathbb{R}^{n \times d}$ with $n \geqslant d$ and whose solution is given in closed-form by $x^* = (A^\top A + \mu I_d)^{-1} A^\top b$. Direct methods yield the exact solution in time $\mathcal{O}(nd^2)$, whereas first-order methods (e.g., conjugate gradient method) yield an δ-approximate solution in time $\mathcal{O}(\sqrt{\kappa} nd \log(1/\delta))$, where κ is the condition number of A. Randomized pre-conditioning and sketching methods can improve on this complexity (see Section 4.1 for further details). Here, our setting for the Newton sketch applies with $f_0(x) = \frac{1}{2}\|Ax - b\|_2^2$ (whose square-root Hessian is A) and $g(x) = \frac{\mu}{2}\|x\|_2^2$ which is μ-strongly convex.

Example 2 (Portfolio optimization)**.** *The optimization problem takes the form*

$$\min_{x \geqslant 0,\, \sum_{i=1}^d x_j \leqslant 1} \left\{ f_0(x) := -r^\top x + \alpha\, \langle x, \Sigma x \rangle \right\}, \tag{28}$$

where $\Sigma = A^\top A$ is an empirical covariance matrix based on the data $A \in \mathbb{R}^{n \times d}$, with $n \geqslant d$. Using the barrier method, we need to solve its penalized version $\min_{x \in \mathbb{R}^d}\{f_0(x) + g(x)\}$, where $g(x) := -\mu \sum_{i=1}^d \log(x_i) - \mu \log(1 - \langle \mathbf{1}, x \rangle)$. We clearly have

the Hessian square-root $\nabla^2 f_0(x)^{1/2} = \sqrt{\alpha}A$. Further, g is μ-strongly convex over its domain: indeed, note that for $0 < x_i < 1$, the Hessian of g is $\mu\,\mathbf{diag}(x_i^2)^{-1} + \mu\mathbf{1}\mathbf{1}^\top$ and the first term satisfies $\mu\,\mathbf{diag}(x_i^2)^{-1} \succeq \mu I_d$.

Example 3 (Solving Lasso via its dual)**.** *Given $A \in \mathbb{R}^{n\times d}$ with $d \gg n$, the dual Lasso problem takes the form*

$$\max_{\|A^T x\|_\infty \leqslant \lambda} \left\{ -\frac{1}{2}\|y - x\|_2^2 \right\}. \tag{29}$$

Applying the logarithmic barrier method, one needs to solve a sequence of problems of the form $\min_{x\in\mathbb{R}^n}\{f_0(x)+g(x)\}$ where $g(x) := \frac{\mu}{2}\|y-x\|_2^2$, $f_0(x) := -\sum_{j=1}^d \log(\lambda - \langle a_j, x\rangle) - \sum_{j=1}^d \log(\lambda + \langle a_j, x\rangle)$ and a_j is the j-th column of A. This form is amenable to the Newton sketch: a square-root of $\nabla^2 f_0(x)$ is given by $\nabla^2 f_0(x)^{1/2} = \mathbf{diag}(|\lambda - \langle a_j, x\rangle|^{-1} + |\lambda + \langle a_j, x\rangle|^{-1})\,A^T$, and the function $g(x)$ is μ-strongly convex.

Example 4 (Regularized logistic regression with $n \gg d$)**.** *We consider data points $\{(a_i, y_i)\}_{i=1}^n$ where each a_i is a d-dimensional feature vector with binary response $y_i \in \{\pm 1\}$. We aim to find a linear classifier through regularized logistic regression, that is,*

$$\min_{x\in\mathbb{R}^d} \left\{ \sum_{i=1}^n \log\left(1 + e^{-y_i a_i^\top x}\right) + \frac{\mu}{2}\|x\|_2^2 \right\}. \tag{30}$$

Setting $f_0(x) = \sum_{i=1}^n \log\left(1 + e^{-y_i a_i^\top x}\right)$ and $g(x) = \frac{\mu}{2}\|x\|_2^2$, we have that $\nabla^2 f_0(x)^{\frac{1}{2}} = \mathbf{diag}(h)A$ where the i-th coefficient of $h \in \mathbb{R}^n$ is given by $h_i = \frac{e^{y_i a_i^\top x/2}}{1+e^{y_i a_i^\top x}}$. More generally, empirical risk minimization with generalized linear models yields a Hessian square-root of the form 'diagonal times data matrix A'.

Example 5 (Projection onto polyhedra)**.** *Given $v \in \mathbb{R}^d$, $A \in \mathbb{R}^{n\times d}$ with $n \gg d$ and $b \in \mathbb{R}^n$ such that there exists $x_0 \in \mathbb{R}^d$ that satisfies $Ax_0 < b$, we aim to solve the optimization problem*

$$\min_{x\in\mathbb{R}^d} \frac{1}{2}\|x - v\|_2^2, \quad \textit{s.t.} \quad Ax \leqslant b. \tag{31}$$

Applying a barrier method, one needs to solve a sequence of optimization problems of the form $\min_x f_0(x) + g(x)$, where $f_0(x) := -\sum_{i=1}^n \log(b_i - a_i^\top x)$ and $g(x) = \frac{\mu}{2}\|x - v\|_2^2$. Clearly, g is μ-strongly convex, and a square-root of $\nabla^2 f_0(x)$ is given by $\mathbf{diag}(|b_i - a_i^\top x|^{-1})A$.

4. Adaptive Newton Sketch with effective dimensionality

We turn to the adaptive version of Algorithm 1, which starts with small sketch size and does not require knowledge or estimation of the effective dimension $\overline{d}_\mu$. Importantly, our method is guaranteed to converge at a tunable geometric rate, and with a sketch size scaling in terms of $\overline{d}_\mu$.

For $\tau \in [0,1]$ and $\varepsilon \in (0, 1/4)$, we set

$$\alpha(\tau,\varepsilon) := \frac{(1+\varepsilon)^{\frac{1}{2}}}{(1-\varepsilon)^{\frac{1+\tau}{2}}}\,\alpha(\tau), \tag{32}$$

and we will consider in this section the sufficient number of iterations $T_{\tau,\alpha(\tau,\varepsilon),\frac{\delta}{d}}$ as defined in Lemma 4 for $\alpha = \alpha(\tau,\varepsilon)$. Our adaptive method is formally described in Algorithm 2. It starts each iteration by checking whether $\widetilde{\lambda}_f(x_t) > \eta$. If so, assuming the sketch size m_t large enough, we have w.h.p. by Lemma 2 that $f(x_{\mathrm{nsk}}) - f(x) \leqslant -\nu$ and we set $x_{t+1} = x_{\mathrm{nsk}}$. Otherwise, if $\widetilde{\lambda}_f(x_t) \leqslant \eta$, we have w.h.p. by Lemma 3 a condition similar to $\widetilde{\lambda}_f(x_{\mathrm{nsk}}) \leqslant \alpha(\tau,\varepsilon)(\widetilde{\lambda}_f(x_t))^{1+\tau}$, in which case we set $x_{t+1} = x_{\mathrm{nsk}}$. If none of the above events happen, we increase the sketch size by a factor 2. On the other hand, if the sketch size is not large enough for the guarantees of Lemmas 2 and 3 to hold w.h.p., then either the algorithm terminates with a potentially small sketch size, or, the sketch size must at some point become large enough due to the doubling trick.

Note that if Algorithm 2 terminates, then it returns an iterate x such that $\widetilde{\lambda}_f(x)^2 \leqslant \frac{\delta}{d}$. We prove next (see Lemma 5) that this termination condition implies the δ-approximation guarantee, i.e., $f(x) - f(x^*) \leqslant \delta$ w.h.p., provided that the *initial* sketch size is large enough, and regardless of the final sketch size.

Lemma 5 (Termination condition)**.** *Let $\delta \in (0, 1/2)$ and $p \in (0, 1/2)$, and suppose that Algorithm 2 returns x. Then, it holds that $f(x) - f(x^*) \leqslant \delta$ with probability at least $1 - p$ provided that $m_0 \gtrsim \log^2(1/p)$ for the SRHT, and, $m_0 \gtrsim 1/p$ for the SJLT.*

4.1. Time and memory space complexity guarantees

For conciseness, we present a succinct version of our complexity guarantees for the adaptive Newton sketch (only for the linear rate $\tau = 0$ and for the quadratic rate $\tau = 1$). A more general statement for any $\tau \in [0,1]$ can be found in the proof of Theorem 3.

Theorem 3 (Geometric convergence guarantees of the adaptive Newton sketch)**.** *Let $\tau \in [0,1]$, $p_0 \in (0, 1/2)$ and $\delta \in (0, 1/2)$. Let $\overline{m}_0$ be an initial sketch size. Then, it holds with probability at least $1 - p_0$ that Algorithm 2 returns a δ-approximate solution $\widetilde{x}$ in function value (i.e., $f(\widetilde{x}) - f(x^*) \leqslant \delta$) in less than $\overline{T} = \mathcal{O}\left(T_{\tau,\alpha(\tau,\varepsilon),\frac{\delta}{d}} \log(\overline{d}_\mu)\right)$ iterations, with final sketch size bounded by $2\,\overline{m}$ and with total time complexity $\overline{C}$. The values of $\overline{m}_0$, $\overline{m}$ and $\overline{C}$ depend on the choice S as follows.*

(SRHT). *For $\tau = 1$ (quadratic rate), picking $p_0 \asymp \frac{\delta}{d}$*

Algorithm 2 Adaptive effective dimension Newton sketch

Require: Initial point $x_0 \in \mathbf{dom} f$, initial sketch size $m_0 = \overline{m}_0$, line-search parameters (a, b), target accuracy $\delta \in (0, 1/2)$, convergence rate parameter $\tau \in [0, 1]$ and sampling precision parameter $\varepsilon = 1/8$.

1: **for** $t = 0, \dots$ **do**
2: Sample $S_t \in \mathbb{R}^{m_t \times n}$ independent of $S_{t-1}, \dots, S_0$.
3: Compute v_{nsk} and $\widetilde{\lambda}_f(x_t)$ based on S_t.
4: **if** $\widetilde{\lambda}_f(x_t)^2 \leqslant \frac{\delta}{d}$ then **return** x_t.
5: Find step size s with backtracking line search, and set $x_{\mathrm{nsk}} = x_t + s v_{\mathrm{nsk}}$.
6: **if** $\widetilde{\lambda}_f(x_t) > \eta$ **then**
7: **if** $f(x_{\mathrm{nsk}}) - f(x) \leqslant -\nu$ **then**
8: Set $x_{t+1} = x_{\mathrm{nsk}}$ and $m_{t+1} = m_t$.
9: **else**
10: Set $x_{t+1} = x_t$ and $m_{t+1} = 2m_t$.
11: **end if**
12: **else**
13: Sample $S^+ \in \mathbb{R}^{m_t \times n}$ independent of $S_t, \dots, S_0$.
14: Compute $v^+ = -H_{S^+}^{-1} \nabla f(x_{\mathrm{nsk}})$ and $\widetilde{\lambda}_f(x_{\mathrm{nsk}}) = (-\langle \nabla f(x_{\mathrm{nsk}}), v^+ \rangle)^{1/2}$.
15: **if** $\widetilde{\lambda}_f(x_{\mathrm{nsk}}) \leqslant \alpha(\tau, \varepsilon)(\widetilde{\lambda}_f(x_t))^{1+\tau}$ **then**
16: Set $x_{t+1} = x_{\mathrm{nsk}}$, $m_{t+1} = m_t$, $v_{\mathrm{nsk}} = v^+$ and go to step 4.
17: **else**
18: Set $x_{t+1} = x_t$ and $m_{t+1} = 2m_t$.
19: **end if**
20: **end if**
21: **end for**

and assuming n large enough such that $n \gtrsim \frac{d^2 \overline{d}_\mu^2}{\delta^2}$, we have $\overline{m}_0 \asymp \frac{d}{\delta} \log(\frac{d}{\delta})$, $\overline{m} \asymp \frac{d}{\delta}(\overline{d}_\mu + \log(\frac{d}{\delta}) \log(\overline{d}_\mu))$, $\overline{T} = \mathcal{O}(\log(\overline{d}_\mu) \log\log(d/\delta))$ and

$$\overline{\mathcal{C}} = \mathcal{O}\Big(nd \log(\overline{d}_\mu) \log(d/\delta) \log\log(d/\delta)\Big) . \tag{33}$$

For $\tau = 0$ (linear rate), picking $p_0 \asymp 1/\overline{d}_\mu$ and assuming n large enough such that $n \gtrsim \frac{\overline{d}_\mu^2}{\log(\overline{d}_\mu)}$, we have $\overline{m}_0 \asymp \log^2(d/\delta)$, $\overline{m} \asymp \overline{d}_\mu$, $\overline{T} = \mathcal{O}(\log(\overline{d}_\mu) \log(d/\delta))$ and

$$\overline{\mathcal{C}} = \mathcal{O}\Big(nd \log^2(\overline{d}_\mu) \log(d/\delta)\Big) . \tag{34}$$

(SJLT). *For $\tau = 1$ (quadratic rate), assuming n large enough such that $n \gtrsim \frac{\overline{d}_\mu^4 d^2 \log(\log(d/\delta))^2}{\delta^2 p_0^2}$, we have $\overline{m}_0 \asymp \frac{d \log(\log(d/\delta))}{p_0 \delta}$, $\overline{m} \asymp \frac{d \overline{d}_\mu^2 \log(\log(d/\delta))}{p_0 \delta}$, $\overline{T} = \mathcal{O}(\log(\overline{d}_\mu) \log\log(d/\delta))$ and*

$$\overline{\mathcal{C}} = \mathcal{O}\big(nd \log(\overline{d}_\mu) \log(\log(d/\delta))\big) . \tag{35}$$

For $\tau = 0$ (linear rate), assuming n large enough such that $n \gtrsim \frac{\overline{d}_\mu^4 \log(d/\delta)^2}{p_0^2}$, we have $\overline{m}_0 \asymp \frac{\log(d/\delta)}{p_0}$, $\overline{m} \asymp \frac{\overline{d}_\mu^2 \log(d/\delta)}{p_0}$, $\overline{T} = \mathcal{O}(\log(\overline{d}_\mu) \log(d/\delta))$ and

$$\overline{\mathcal{C}} = \mathcal{O}\big(nd \log(\overline{d}_\mu) \log(\log(d/\delta))\big) . \tag{36}$$

Note that adaptivity with convergence rate parameter τ comes at the cost of an additional d^τ factor for the final sketch size, compared to Algorithm 1. This is essentially due to our exit condition threshold δ/d that we choose for the following reason. For small $m \gtrsim 1$, the approximate Newton decrement $\widetilde{\lambda}_f^2(x)$ may fluctuate around $\lambda_f^2(x)$ by a factor up to $\overline{d}_\mu$ (see Theorem 1 in (Cohen et al., 2015)). In this case, the exit condition $\widetilde{\lambda}_f(x_t)^2 \approx \delta$ would result in $f(x_t) - f(x^*) \approx \delta \overline{d}_\mu$. To guarantee δ-accuracy, it is sufficient to use the termination condition $\delta/\overline{d}_\mu$ to account for these fluctuations. As $\overline{d}_\mu$ is unknown, we choose to divide instead by d.

We summarize our complexity guarantees in Table 1. In contrast to gradient descent (GD), Nesterov's accelerated gradient descent (NAG) and Newton's method (NE), our time complexity has no condition number dependency and scales linearly in nd up to log-factors, and so does the original Newton sketch (NS). The NS log-factor is at least $\log(d)\log(1/\delta)$ whereas our SRHT-quadratic mode adaptive method has a log-factor $\log(\overline{d}_\mu)\log(d/\delta)\log(\log(d/\delta))$. The latter is much smaller for effective dimension $\overline{d}_\mu$ small compared to d and $1/\delta$. Furthermore, in terms of memory, our algorithm starts with small m whereas NS uses a constant sketch size $m \gtrsim d$. For $\tau = 0$, our memory savings are drastic when $\overline{d}_\mu$ is small. There are downsides to our method, in comparison to NS. For $\overline{d}_\mu$ close to d, our time complexity bounds become worse than NS, by an adaptivity-cost factor $\log \overline{d}_\mu$ for both $\tau = 0$ and $\tau = 1$. For $\tau = 1$, our *worst-case* sketch size is always greater than that of NS, by a factor $\overline{d}_\mu/\delta$, which comes from enforcing quadratic convergence.

When $\log \overline{d}_\mu \gg \log\log(d/\delta)$, then our SRHT/quadratic mode adaptive method yields a better time complexity than its linear mode counterpart, but at the expense of worse memory complexity.

In the context of ridge regression, we note that the time complexity of our SRHT-linear mode adaptive method scales similarly to the complexity of the adaptive method proposed by (Lacotte & Pilanci, 2020) for returning a certified δ-accurate solution. Importantly, our method applies to a much broader range of optimization problems, and can achieve better complexity by tuning the convergence rate parameter τ.

We emphasize again that our guarantees hold in a worst-case sense. In practice, the sketch size can start from a small value and may remain significantly smaller than the bounds in Table 1, which we illustrate in our numerical experiments.

Table 1. We compare the time complexity of different optimization methods in order to achieve a δ-accurate solution in function value, for a function with condition number κ. 'NS-effdim' (resp. 'NS-ada') refers to our Algorithm 1 (resp. our Algorithm 2); 'linear' (resp. 'quadratic') signifies the choice $\tau = 0$ (resp. $\tau = 1$). We refer to (Nesterov, 2003) for gradient descent (GD), Nesterov's accelerated gradient method (NAG) and Newton's method (NE); we refer to (Pilanci & Wainwright, 2017) for the Newton sketch (NS), and we refer to Theorems 2 and 3 for our algorithms. Katyusha (Allen-Zhu, 2017) is an instance of an accelerated SVRG method whose condition number dependency improves on NAG. For each algorithm, we assume that the sample size n is large enough for the time complexity to scale at least linearly in the term nd.

Algorithm	Time complexity	Sketch size	Proba.	Linear scaling regime
GD	$\kappa\, nd\, \log(1/\delta)$	-	1	-
NAG	$\sqrt{\kappa}\, nd\, \log(1/\delta)$	-	1	-
Katyusha	$(nd + d\sqrt{\kappa n})\log(1/\delta)$	-	1	-
NE	$nd^2 \log(\log(1/\delta))$	-	1	-
NS	$nd \log(d) \log(1/\delta)$	d	$1 - \frac{1}{d}$	$n \gtrsim \frac{d^2}{\log d}$
NS-effdim (SHRT, linear)	$nd \log(\overline{d}_\mu) \log(1/\delta)$	$\overline{d}_\mu$	$1 - \frac{1}{\overline{d}_\mu}$	$n \gtrsim \frac{d_\mu^2}{\log(d_\mu)}$
NS-effdim (SHRT, quadratic)	$nd \log(\overline{d}_\mu/\delta) \log(\log(1/\delta))$	$\frac{\overline{d}_\mu}{\delta} \log(\overline{d}_\mu/\delta)$	$1 - \frac{1}{\overline{d}_\mu}$	$n \gtrsim \frac{\overline{d}_\mu^2 \log(\frac{\overline{d}_\mu}{\delta})}{\delta^2}$
NS-ada (SRHT, linear)	$nd \log(\overline{d}_\mu)^2 \log(d/\delta)$	$\overline{d}_\mu$	$1 - \frac{1}{\overline{d}_\mu}$	$n \gtrsim \frac{\overline{d}_\mu^2}{\log(\overline{d}_\mu)}$
NS-ada (SRHT, quadratic)	$nd \log(\overline{d}_\mu) \log(\frac{d}{\delta}) \log(\log(\frac{d}{\delta}))$	$\frac{d}{\delta}\left(\overline{d}_\mu + \log(\frac{d}{\delta}) \log(\overline{d}_\mu)\right)$	$1 - \frac{1}{\overline{d}_\mu}$	$n \gtrsim \frac{d^2 \overline{d}_\mu^2}{\delta^2}$

5. Numerical experiments

In this section, we compare adaptive Newton Sketch (NS-ada) with other optimization methods in regularized logistic regression problems as in Example 4. The compared methods include Newton Sketch (NS) with fixed sketching dimension, Newton's method (NE), gradient descent method (GD), Nesterov's accelerated gradient descent method (NAG) (Nesterov, 1983), the stochastic variance reduced gradient method (SVRG) (Johnson & Zhang, 2013) and Katyusha (Allen-Zhu, 2017) . For NS-ada and NS, we consider both SJLT sketching matrices and random row sampling (RSS) sketching matrices. For SVRG and Katyusha, we use a batch size of 20. All numerical experiments are executed on a Dell PowerEdge R840 workstation. Specifically, we use 4 cores with 192GB ram for all compared methods.

The datasets used in the numerical experiments are collected from LIBSVM[1] (Chih-Chung & Chih-Jen, 2011). The datasets for multi-class classification are manually separated into two categories. For example, in MNIST dataset, we classify even and odd digits. For each dataset, we randomly split it into a training set and a test set with the ratio $1:1$. Several additional numerical results and experimental details are reported in Appendix A.

5.1. Regularized logistic regression

We demonstrate the performance of all compared methods on regularized logistic regression problems. The relative error is calculated by $\frac{f(x)-f_{\text{ref}}+\epsilon}{1+f_{\text{ref}}}$. Here f_{ref} is the minimal training loss function value among all compared methods and $\epsilon = 5 \times 10^{-7}$ is a small constant.

We report the relative error and the test error with respect to the iteration number and the CPU-time in Figures 1 and 2. NS-ada-SJLT or NS-ada-RRS can achieve the best performance in the relative error with respect to the CPU-time. We can also obverse the super-linear asymptotic convergence rate of NS-ada as it gets closer to the optimum. Compared to NS, NS-ada requires less iterations and less time to converge to a solution with small relative error. Compared to methods utilizing second-order information, first-order methods are less competitive to find a high-precision solution.

[1] https://www.csie.ntu.edu.tw/ cjlin/libsvm/

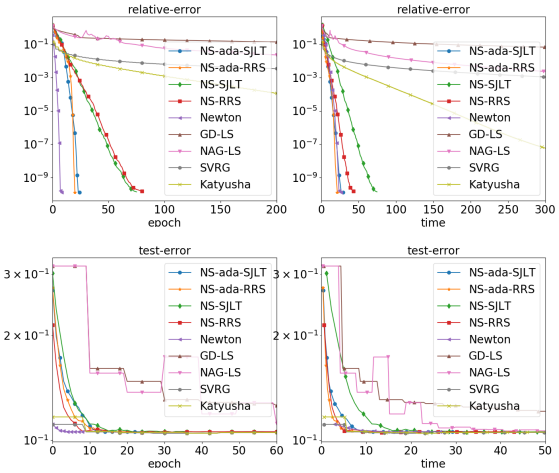

Figure 1. MNIST. $n = 30000, d = 780, \mu = 10^{-1}$.

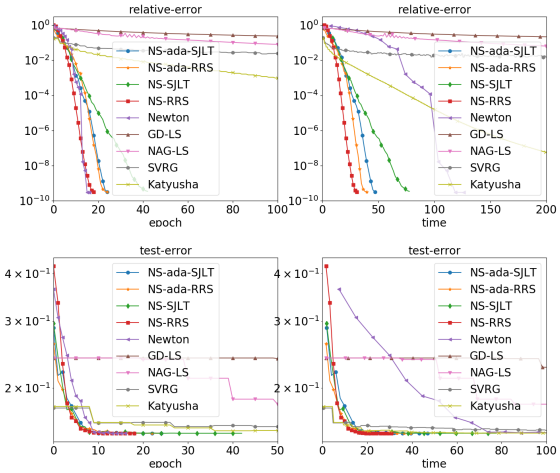

Figure 3. a8a. kernel matrix. $n = 10000, d = 10000, \mu = 10$.

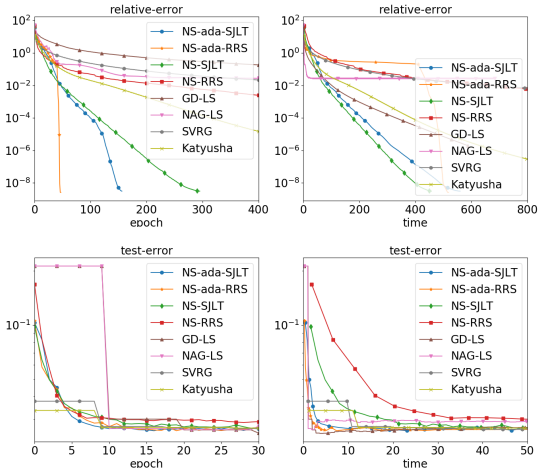

Figure 2. realsim. $n = 50000, d = 20958, \mu = 10^{-3}$.

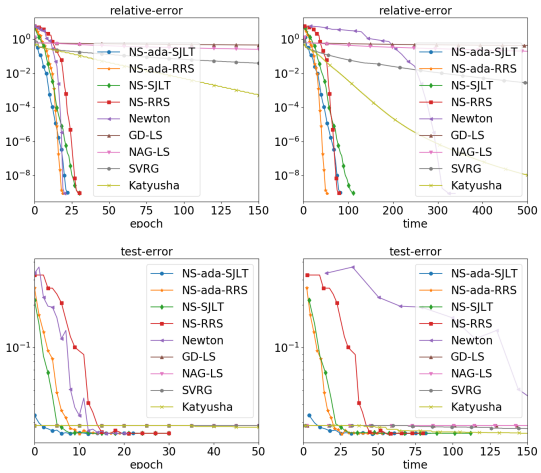

Figure 4. w7a. kernel matrix. $n = 12000, d = 12000, \mu = 10$.

5.2. Regularized logistic regression with kernel matrix

We also test on regularized logistic regression with the kernel matrix. The relative error and the test error with respect to the iteration number and the CPU-time are plotted in Figure 5 to 9. NS-ada-SJLT and NS-ada-RRS demonstrate asymptotic super-linear convergence rate of the relative error as the Newton's method in terms of iteration numbers. They also achieve a rapid decrease in relative error in terms of CPU-time. First-order methods have worst performance in terms of relative error. This may come from that the kernel matrices are usually ill-conditioned, i.e., with large condition number.

Acknowledgements

We would like to thank the reviewers for their comments. This work was partially supported by the National Science Foundation under grants IIS-1838179 and ECCS-2037304, Facebook Research, Adobe Research and Stanford SystemX Alliance.

References

Ailon, N. and Chazelle, B. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pp. 557–563. ACM, 2006.

Alaoui, A. E. and Mahoney, M. W. Fast randomized ker-

nel ridge regression with statistical guarantees. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pp. 775–783, 2015.

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

Asi, H. and Duchi, J. C. The importance of better models in stochastic optimization. *Proceedings of the National Academy of Sciences*, 116(46):22924–22930, 2019.

Avron, H., Maymounkov, P., and Toledo, S. Blendenpik: Supercharging lapack's least-squares solver. *SIAM Journal on Scientific Computing*, 32(3):1217–1236, 2010.

Avron, H., Clarkson, K. L., and Woodruff, D. P. Sharper bounds for regularized data fitting. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2017.

Bach, F. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pp. 185–209. PMLR, 2013.

Bartan, B. and Pilanci, M. Distributed sketching methods for privacy preserving regression. *arXiv preprint arXiv:2002.06538*, 2020.

Berahas, A. S., Bollapragada, R., and Nocedal, J. An investigation of newton-sketch and subsampled newton methods. *Optimization Methods and Software*, 35(4):661–680, 2020.

Bollapragada, R., Byrd, R. H., and Nocedal, J. Exact and inexact subsampled newton methods for optimization. *IMA Journal of Numerical Analysis*, 39(2):545–578, 2019.

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

Byrd, R. H., Chin, G. M., Neveitt, W., and Nocedal, J. On the use of stochastic hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 21(3):977–995, 2011.

Chih-Chung, C. and Chih-Jen, L. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

Cohen, M. B., Nelson, J., and Woodruff, D. P. Optimal approximate matrix product in terms of stable rank. *arXiv preprint arXiv:1507.02268*, 2015.

Derezinski, M., Bartan, B., Pilanci, M., and Mahoney, M. W. Debiasing distributed second order optimization with surrogate sketching and scaled regularization. In *Conference on Neural Information Processing Systems*, 2020.

Dobriban, E. and Liu, S. Asymptotics for sketching in least squares regression. In *Advances in Neural Information Processing Systems*, pp. 3670–3680, 2019.

Doikov, N. and Richtárik, P. Randomized block cubic Newton method. *International Conference on Machine Learning*, 2018.

Drineas, P. and Mahoney, M. W. RandNLA: randomized numerical linear algebra. *Communications of the ACM*, 59(6):80–90, 2016.

Gower, R., Koralev, D., Lieder, F., and Richtárik, P. RSN: Randomized subspace newton. In *Advances in Neural Information Processing Systems*, pp. 616–625, 2019.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013.

Lacotte, J. and Pilanci, M. Effective dimension adaptive sketching methods for faster regularized least-squares optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

Lacotte, J., Pilanci, M., and Pavone, M. High-dimensional optimization in adaptive random subspaces. *arXiv preprint arXiv:1906.11809*, 2019.

Lacotte, J., Liu, S., Dobriban, E., and Pilanci, M. Limiting spectrum of randomized hadamard transform and optimal iterative sketching methods. In *Conference on Neural Information Processing Systems*, 2020.

Li, X., Wang, S., and Zhang, Z. Do subsampled newton methods work for high-dimensional data? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4723–4730, 2020.

Mahoney, M. W. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3 (2):123–224, 2011.

Meng, X., Saunders, M. A., and Mahoney, M. W. Lsrn: A parallel iterative solver for strongly over-or underdetermined systems. *SIAM Journal on Scientific Computing*, 36(2):C95–C118, 2014.

Nelson, J. and Nguyên, H. L. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *2013 ieee 54th annual symposium on foundations of computer science*, pp. 117–126. IEEE, 2013.

Nesterov, Y. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Nesterov, Y. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Nesterov, Y. and Polyak, B. T. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

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

Qu, Z., Richtárik, P., Takác, M., and Fercoq, O. SDNA: Stochastic dual Newton ascent for empirical risk minimization. In *International Conference on Machine Learning*, pp. 1823–1832, 2016.

Rokhlin, V. and Tygert, M. A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences*, 105(36): 13212–13217, 2008.

Roosta-Khorasani, F. and Mahoney, M. W. Sub-sampled newton methods. *Mathematical Programming*, 174(1): 293–326, 2019.

Shamir, O., Srebro, N., and Zhang, T. Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pp. 1000–1008. PMLR, 2014.

Vempala, S. S. *The random projection method*, volume 65. American Mathematical Society, 2005.

Wang, S., Roosta, F., Xu, P., and Mahoney, M. W. Giant: Globally improved approximate newton method for distributed optimization. *Advances in Neural Information Processing Systems*, 31:2332–2342, 2018.

Woodruff, D. P. et al. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Xu, P., Yang, J., Roosta-Khorasani, F., Ré, C., and Mahoney, M. W. Sub-sampled newton methods with non-uniform sampling. *arXiv preprint arXiv:1607.00559*, 2016.

Xu, P., Roosta, F., and Mahoney, M. W. Second-order optimization for non-convex machine learning: An empirical study. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pp. 199–207. SIAM, 2020.

Yang, Y., Pilanci, M., Wainwright, M. J., et al. Randomized sketches for kernels: Fast and optimal nonparametric regression. *The Annals of Statistics*, 45(3):991–1023, 2017.