NEWTON SKETCH: A NEAR LINEAR-TIME OPTIMIZATION ALGORITHM WITH LINEAR-QUADRATIC CONVERGENCE

MERT PILANCI[†] AND MARTIN J. WAINWRIGHT[†] [‡]

Abstract. We propose a randomized second-order method for optimization known as the Newton Sketch: it is based on performing an approximate Newton step using a randomly projected Hessian. For self-concordant functions, we prove that the algorithm has super-linear convergence with exponentially high probability, with convergence and complexity guarantees that are independent of condition numbers and related problem-dependent quantities. Given a suitable initialization, similar guarantees also hold for strongly convex and smooth objectives without self-concordance. When implemented using randomized projections based on a sub-sampled Hadamard basis, the algorithm typically has substantially lower complexity than Newton's method. We also describe extensions of our methods to programs involving convex constraints that are equipped with self-concordant barriers. We discuss and illustrate applications to linear programs, quadratic programs with convex constraints, logistic regression and other generalized linear models, as well as semidefinite programs.

Key words. Convex optimization, large-scale problems, Newton's Method, random projection, randomized algorithms, random matrices, self-concordant functions, interior point method

AMS subject classifications. 49M15, 90C06, 90C25, 90C51, 62J12

1. Introduction.

Relative to first-order methods, second-order methods for convex optimization enjoy superior convergence in both theory and practice. For instance, Newton's method converges at a quadratic rate for strongly convex and smooth problems. Even for functions that are weakly convex—that is, convex but not strongly convex—modifications of Newton's method have super-linear convergence (for instance, see the paper [39] for an analysis of the Levenberg-Marquardt Method). This rate is faster than the $1/T^2$ convergence rate that can be achieved by a first-order method like accelerated gradient descent, with the latter rate known to be unimprovable (in general) for first-order methods [27]. Yet another issue in first-order methods is the tuning of step size, whose optimal choice depends on the strong convexity parameter and/or smoothness of the underlying problem. For example, consider the problem of optimizing a function of the form $x \mapsto g(Ax)$, where $A \in \mathbb{R}^{n \times d}$ is a "data matrix", and $g : \mathbb{R}^n \to \mathbb{R}$ is a twice-differentiable function. Here the performance of first-order methods will depend on both the convexity/smoothness of g, as well as the conditioning of the data matrix. In contrast, whenever the function g is self-concordant, then Newton's method with suitably damped steps has a global complexity guarantee that is provably independent of such problem-dependent parameters.

On the other hand, each step of Newton's method requires solving a linear system defined by the Hessian matrix. For instance, in application to the problem family just described involving an $n \times d$ data matrix, each of these steps has complexity scaling as $\mathcal{O}(nd^2)$ assuming $n \geq d$ without loss of generality. For this reason, both forming the Hessian and solving the corresponding linear system pose a tremendous numerical challenge for large values of (n, d)— for instance, values of thousands to millions, as is common in big data applications. In order to address this issue, a wide variety of different approximations to Newton's method have been proposed and studied. The general class of quasi-Newton methods are based on estimating the inverse Hessian using successive evaluations of the gradient vectors. Examples of such quasi-Newton methods include DFP and BFGS schemes as well their limited memory versions; see the book by Wright and Nocedal [38] and references therein for further details. A disadvantage of such first-order Hessian approximations is that the associated convergence guarantees are typically weaker than those of Newton's method and require stronger assumptions.

In this paper, we propose and analyze a randomized approximation of Newton's method, known as the *Newton Sketch*. Instead of explicitly computing the Hessian, the Newton Sketch method approximates it via a random projection of dimension m. When these projections are carried out using the fast Johnson-Lindenstrauss (JL) transform, say based on Hadamard matrices, each iteration has complexity $\mathcal{O}(nd \log(m) + dm^2)$. Our results show that it is always sufficient to choose m proportional to $\min\{d, n\}$, and moreover, that the sketch dimension m can be much smaller for certain types of

[†] Department of Electrical Engineering and Computer Science, University of California, Berkeley.

[‡] Department of Statistics, University of California, Berkeley.

constrained problems. Thus, in the regime $n > d$ and with $m \asymp d$, the complexity per iteration can be substantially lower than the $\mathcal{O}(nd^2)$ complexity of each Newton step. For instance, for an objective function of the form $f(x) = g(Ax)$ in the regime $n \geq d^2$, the complexity of Newton Sketch per iteration is $\mathcal{O}(nd \log d)$, which (modulo the logarithm) is linear in the input data size nd. Thus, the computational complexity per iteration is comparable to first-order methods that have access only to the gradient $A^T g'(Ax)$. In contrast to first-order methods, we show that for self-concordant functions, the total complexity of obtaining a δ-approximate solution is $\mathcal{O}\big(nd(\log d)\log(1/\delta)\big)$, and without any dependence on constants such as strong convexity or smoothness parameters. Moreover, for problems with $d > n$, we provide a dual strategy that effectively has the same guarantees with roles of d and n exchanged.

We also consider other random projection matrices and sub-sampling strategies, including partial forms of random projection that exploit known structure in the Hessian. For self-concordant functions, we provide an affine invariant analysis proving that the convergence is linear-quadratic and the guarantees are independent of various problem parameters, such as condition numbers of matrices involved in the objective function. Finally, we describe an interior point method to deal with arbitrary convex constraints, which combines the Newton sketch with the barrier method. We provide an upper bound on the total number of iterations required to obtain a solution with a pre-specified target accuracy.

The remainder of this paper is organized as follows. We begin in Section 2 with some background on the classical form of Newton's method, past work on approximate forms of Newton's method, random matrices for sketching, and Gaussian widths as a measure of the size of a set. In Section 3, we formally introduce the Newton Sketch, including both fully and partially sketched versions for unconstrained and constrained problems. We provide some illustrative examples in Section 3.3 before turning to local convergence theory in Section 3.4. Section 4 is devoted to global convergence results for self-concordant functions, in both the constrained and unconstrained settings. In Section 5, we consider a number of applications and provide additional numerical results. The bulk of our proofs are in given in Section 6, with some more technical aspects deferred to the appendices.

2. Background.

We begin with some background material on the standard form of Newton's method, past work on approximate or stochastic forms of Newton's method, the basics of random sketching, and the notion of Gaussian width as a complexity measure.

2.1. Classical version of Newton's method.

In this section, we briefly review the convergence properties and complexity of the classical form of Newton's method; see the sources [38, 6, 27] for further background. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a closed, convex and twice-differentiable function that is bounded below. Given a convex and closed set $\mathcal{C}$, we assume that the constrained minimizer

$$x^* := \arg\min_{x \in \mathcal{C}} f(x) \tag{2.1}$$

exists and is uniquely defined. We define the minimum and maximum eigenvalues $\gamma = \lambda_{min}(\nabla^2 f(x^*))$ and $\beta = \lambda_{max}(\nabla^2 f(x^*))$ of the Hessian evaluated at the minimum.

We assume moreover that the Hessian map $x \mapsto \nabla^2 f(x)$ is Lipschitz continuous with modulus L, meaning that

$$|\!|\!| \nabla^2 f(x + \Delta) - \nabla^2 f(x) |\!|\!|_{\mathrm{op}} \leq L \, \|\Delta\|_2. \tag{2.2}$$

Under these conditions and given an initial point $\tilde{x}^0 \in \mathcal{C}$ such that $\|\tilde{x}^0 - x^*\|_2 \leq \frac{\gamma}{2L}$, the Newton updates are guaranteed to converge quadratically—viz.

$$\|\tilde{x}^{t+1} - x^*\|_2 \leq \frac{2L}{\gamma} \|\tilde{x}^t - x^*\|_2^2,$$

This result is classical: for instance, see Boyd and Vandenberghe [6] for a proof. Newton's method can be slightly modified to be globally convergent by choosing the step sizes via a simple backtracking line-search procedure.

The following result characterizes the complexity of Newton's method when applied to self-concordant functions and is central in the development of interior point methods (for instance, see the books [28, 6]). We defer the definitions of self-concordance and the line-search procedure to the following sections. The number of iterations needed to obtain a δ-approximate minimizer of a strictly convex self-concordant function f is at most

$$\frac{20-8a}{ab(1-2a)}\big(f(x^0)-f(x^*)\big)+\log_2\log_2(1/\delta)\,,$$

where a, b are constants in the line-search procedure.[*]

2.2. Approximate Newton methods. Given the complexity of the exact Newton updates, various forms of approximate and stochastic variants of Newton's method have been proposed, which we discuss here. In general, inexact solutions of the Newton updates can be used to guarantee convergence while reducing overall computational complexity [11, 12]. In the unconstrained setting, the Newton update corresponds to solving a linear system of equations, and one approximate approach is truncated Newton's method: it involves applying the conjugate gradient (CG) method for a specified number of iterations, and then using the solution as an approximate Newton step [12]. In applying this method, the Hessian need not be formed since the CG updates only need access to matrix-vector products with the Hessian. These matrix vector products can also be approximated using finite differences of gradients (e.g., see [23]). While these strategies are popular, theoretical analysis of inexact Newton methods typically need strong assumptions on the eigenvalues of the Hessian [11]. Since the number of steps of CG for reaching a certain residual error necessarily depends on the condition number, the overall complexity of truncated Newton's Method is problem-dependent; the condition numbers can be arbitrarily large, and in general are unknown *a priori*. Ill-conditioned Hessian system are common in applications of Newton's method within interior point methods. Consequently, software toolboxes typically perform approximate Newton steps using CG updates in earlier iterations, but then shift to exact Newton steps via Cholesky or QR decompositions in later iterations.

A more recent line of work, inspired by the success of stochastic first-order algorithms for large scale machine learning applications, has focused on stochastic forms of second-order optimization algorithms (e.g., [33, 5, 7, 8]). Schraudolph et al. [33] use online limited memory BFGS-like updates to maintain an inverse Hessian approximation. Byrd et al. [8, 7] propose stochastic second-order methods that use batch sub-sampling in order to obtain curvature information in a computationally inexpensive manner. These methods are numerically effective in problems in which objective consists of a sum of a large number of individual terms; however, their theoretical analysis again involves strong assumptions on the eigenvalues of the Hessian. Moreover, such second-order methods do not retain the affine invariance of the original Newton's method, which guarantees iterates are independent of the coordinate system and conditioning. When simple stochastic schemes like sub-sampling are used to approximate the Hessian, affine invariance is lost, since subsampling is coordinate and conditioning dependent. In contrast, the stochastic form of Newton's method analyzed in this paper is constructed so as to retain this affine invariance property, and thus not depend on the problem conditioning.

2.3. Different types of randomized sketches. Our Newton sketch algorithm is based on performing a form of dimensionality reduction using random matrices, known as *sketching matrices*. Various types of randomized sketches are possible, and we describe a few of them here. Given a sketching matrix $S \in \mathbb{R}^{m\times n}$, we use $\{s_i\}_{i=1}^m$ to denote the collection of its n-dimensional rows. We restrict our attention to sketch matrices that are zero-mean, and that are normalized so that $\mathbb{E}[S^TS/m] = I_n$.

Sub-Gaussian sketches. The most classical sketch is based on a random matrix $S \in \mathbb{R}^{m\times n}$ with i.i.d. standard Gaussian entries, or somewhat more generally, sketch matrices based on i.i.d. sub-Gaussian rows. In particular, a zero-mean random vector $s \in \mathbb{R}^n$ is 1-sub-Gaussian if for any $u \in \mathbb{R}^n$, we have

$$\mathbb{P}[\langle s,\, u\rangle \geq \epsilon\|u\|_2\big] \leq e^{-\epsilon^2/2} \qquad \text{for all } \epsilon \geq 0. \tag{2.3}$$

For instance, a vector with i.i.d. $N(0,1)$ entries is 1-sub-Gaussian, as is a vector with i.i.d. Rademacher entries (uniformly distributed over $\{-1,+1\}$). We use the terminology *sub-Gaussian sketch* to mean a random matrix $S \in \mathbb{R}^{m\times n}$ with i.i.d. rows that are zero-mean, 1-sub-Gaussian, and with $\text{cov}(s) = I_n$.

From a theoretical perspective, sub-Gaussian sketches are attractive because of the well-known concentration properties of sub-Gaussian random matrices (e.g., [10, 37]). On the other hand, from a

[*] Typical values of these constants are $a = 0.1$ and $b = 0.5$ in practice.

computational perspective, a disadvantage of sub-Gaussian sketches is that they require matrix-vector multiplications with unstructured random matrices. In particular, given a data matrix $A \in \mathbb{R}^{n\times d}$, computing its sketched version SA requires $\mathcal{O}(mnd)$ basic operations in general (using classical matrix multiplication).

Sketches based on randomized orthonormal systems (ROS). The second type of randomized sketch we consider is *randomized orthonormal system* (ROS), for which matrix multiplication can be performed much more efficiently. In order to define a ROS sketch, we first let $H \in \mathbb{C}^{n\times n}$ be an orthonormal complex valued matrix with unit magnitude entries, i.e., $|H_{ij}| \in [-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$. Standard classes of such matrices are the Hadamard or Fourier bases, for which matrix-vector multiplication can be performed in $\mathcal{O}(n \log n)$ time via the fast Hadamard or Fourier transforms, respectively. Based on any such matrix, a sketching matrix $S \in \mathbb{C}^{m\times n}$ from a ROS ensemble is obtained by sampling i.i.d. rows of the form

$$s^T = \sqrt{n} e_j^T H D \qquad \text{with probability } 1/n \text{ for } j = 1, \ldots, n,$$

where the random vector $e_j \in \mathbb{R}^n$ is chosen uniformly at random from the set of all n canonical basis vectors, and $D = \mathrm{diag}(\nu)$ is a diagonal matrix of i.i.d. Rademacher variables $\nu \in \{-1, +1\}^n$. Given a fast routine for matrix-vector multiplication, the sketch SM for a data matrix $M \in \mathbb{R}^{n\times d}$ can be formed in $\mathcal{O}(n\, d \log m)$ time (for instance, see the papers [3, 2, 14]). The fast matrix multiplication usually requires n to be a power of 2 (or power of r for a radix-r construction). However, in order to use the fast multiplication for an arbitrary n, we can augment the data matrix with a block of zero rows and do the same for the square root of the Hessian without changing the objective value.

Sketches based on random row sampling. Given a probability distribution $\{p_j\}_{j=1}^n$ over $[n] = \{1, \ldots, n\}$, another choice of sketch is to randomly sample the rows of a data matrix M a total of m times with replacement from the given probability distribution. Thus, the rows of S are independent and take on the values

$$s^T = \frac{e_j}{\sqrt{p_j}} \qquad \text{with probability } p_j \text{ for } j = 1, \ldots, n$$

where $e_j \in \mathbb{R}^n$ is the j^{th} canonical basis vector. Different choices of the weights $\{p_j\}_{j=1}^n$ are possible, including those based on the row ℓ_2 norms $p_j \propto \|Me_j\|_2^2$ and leverage values of M—i.e., $p_j \propto \|Ue_j\|_2$ for $j = 1, \ldots, n$, where $U \in \mathbb{R}^{n\times d}$ is the matrix of left singular vectors of M (e.g., see the paper [13]). When the matrix $M \in \mathbb{R}^{n\times d}$ corresponds to the adjacency matrix of a graph with d vertices and n edges, the leverage scores of M are also known as effective resistances which can be used to sub-sample edges of a given graph by preserving its spectral properties [35].

Sparse JL Sketches. For sparse data matrices, the sketching operation can be done faster if the sketching matrix is chosen from a distribution over sparse matrices. Several works developed sparse JL embeddings [1, 9, 19] and sparse subspace embeddings [25]. Here we describe a construction given by [25, 19]. Given an integer s, each column of S is chosen to have exactly s non-zero entries in random locations, each equal to $\pm 1/\sqrt{s}$ uniformly at random. The column sparsity parameter s can be chosen $O(1/\epsilon)$ for subspace embeddings and $O(\log(1/\delta)/\epsilon)$ for sparse JL embeddings where δ is the failure probability.

2.4. Gaussian widths.

In this section, we introduce some background on the notion of Gaussian width, a way of measuring the size of a compact set in $\mathbb{R}^d$. These width measures play a key role in the analysis of randomized sketches. Given a compact subset $\mathcal{L} \subseteq \mathbb{R}^d$, its Gaussian width is given by

$$\mathcal{W}(\mathcal{L}) := \mathbb{E}_g\big[\max_{z\in\mathcal{L}} |\langle g,\, z\rangle|\big] \tag{2.4}$$

where $g \in \mathbb{R}^n$ is an i.i.d. sequence of $N(0, 1)$ variables. This complexity measure plays an important role in Banach space theory, learning theory and statistics (e.g., [32, 21, 4]).

Of particular interest in this paper are sets $\mathcal{L}$ that are obtained by intersecting a given cone $\mathcal{K}$ with the Euclidean sphere $\mathcal{S}^{d-1} = \{z \in \mathbb{R}^n \mid \|z\|_2 = 1\}$. It is easy to show that the Gaussian width of any such set is at most $\sqrt{d}$, but the it can be substantially smaller, depending on the nature of the underlying cone. For instance, if $\mathcal{K}$ is a subspace of dimension $r < d$, then a simple calculation yields that $\mathcal{W}(\mathcal{K} \cap \mathcal{S}^{d-1}) \leq \sqrt{r}$.

3. Newton sketch and local convergence. With the basic background in place, let us now introduce the Newton sketch algorithm, and then develop a number of convergence guarantees associated with it. It applies to an optimization problem of the form $\min_{x\in\mathcal{C}} f(x)$, where $f : \mathbb{R}^d \to \mathbb{R}$ is a twice-differentiable convex function, and $\mathcal{C} \subseteq \mathbb{R}^d$ is a closed and convex constraint set.

3.1. Newton sketch algorithm. In order to motivate the Newton sketch algorithm, recall the standard form of Newton's algorithm: given a current iterate $\tilde{x}^t \in \mathcal{C}$, it generates the new iterate $\tilde{x}^{t+1}$ by performing a constrained minimization of the second order Taylor expansion—viz.

$$\tilde{x}^{t+1} = \arg\min_{x\in\mathcal{C}} \Big\{ \frac{1}{2}\langle x - \tilde{x}^t,\, \nabla^2 f(\tilde{x}^t)\,(x-\tilde{x}^t)\rangle + \langle \nabla f(\tilde{x}^t),\, x - \tilde{x}^t\rangle \Big\}. \tag{3.1a}$$

In the unconstrained case—that is, when $\mathcal{C} = \mathbb{R}^d$—it takes the simpler form

$$\tilde{x}^{t+1} = \tilde{x}^t - \big[\nabla^2 f(\tilde{x}^t)\big]^{-1} \nabla f(\tilde{x}^t)\,. \tag{3.1b}$$

Now suppose that we have available a Hessian matrix square root $\nabla^2 f(x)^{1/2}$—that is, a matrix $\nabla^2 f(x)^{1/2}$ of dimensions $n \times d$ such that

$$(\nabla^2 f(x)^{1/2})^T \nabla^2 f(x)^{1/2} = \nabla^2 f(x) \qquad \text{for some integer } n \geq \operatorname{rank}(\nabla^2 f(x)).$$

In many cases, such a matrix square root can be computed efficiently. For instance, consider a function of the form $f(x) = g(Ax)$ where $A \in \mathbb{R}^{n\times d}$, and the function $g : \mathbb{R}^n \to \mathbb{R}$ has the separable form $g(Ax) = \sum_{i=1}^n g_i(\langle a_i,\, x\rangle)$. In this case, a suitable Hessian matrix square root is given by the $n\times d$ matrix $\nabla^2 f(x)^{1/2} := \operatorname{diag}\left\{ g_i''(\langle a_i,\, x\rangle)^{1/2}\right\}_{i=1}^n A$. In Section 3.3, we discuss various concrete instantiations of such functions.

In terms of this notation, the ordinary Newton update can be re-written as

$$\tilde{x}^{t+1} = \arg\min_{x\in\mathcal{C}} \Big\{ \underbrace{\frac{1}{2}\|\nabla^2 f(\tilde{x}^t)^{1/2}(x-\tilde{x}^t)\|_2^2 + \langle \nabla f(\tilde{x}^t),\, x-\tilde{x}^t\rangle}_{\tilde{\Phi}(x)} \Big\},$$

and the Newton Sketch algorithm is most easily understood based on this form of the updates. More precisely, for a sketch dimension m to be chosen, let $S \in \mathbb{R}^{m\times n}$ be a sub-Gaussian, ROS, sparse-JL sketch or subspace embedding (when $\mathcal{C}$ is a subspace), satisfying the relation $\mathbb{E}[S^T S] = I_n$. The *Newton Sketch algorithm* generates a sequence of iterates $\{x^t\}_{t=0}^\infty$ according to the recursion

$$x^{t+1} \in \arg\min_{x\in\mathcal{C}} \Big\{ \underbrace{\frac{1}{2}\|S^t \nabla^2 f(x^t)^{1/2}(x-x^t)\|_2^2 + \langle \nabla f(x^t),\, x-x^t\rangle}_{\Phi(x;S^t)} \Big\}, \tag{3.2}$$

where $S^t \in \mathbb{R}^{m\times d}$ is an independent realization of a sketching matrix. When the problem is unconstrained, i.e., $\mathcal{C} = \mathbb{R}^d$ and the matrix $\nabla^2 f(x^t)^{1/2}(S^t)^T S^t \nabla^2 f(x^t)^{1/2}$ is invertible, the Newton sketch update takes the simpler form

$$x^{t+1} = x^t - \Big(\nabla^2 f(x^t)^{1/2}(S^t)^T S^t \nabla^2 f(x^t)^{1/2}\Big)^{-1} \nabla f(x^t). \tag{3.3}$$

The intuition underlying the Newton sketch updates is as follows: the iterate x^{t+1} corresponds to the constrained minimizer of the random objective function $\Phi(x;S^t)$ whose expectation $\mathbb{E}[\Phi(x;S^t)]$, taking averages over the isotropic sketch matrix S^t, is equal to the original Newton objective $\tilde{\Phi}(x)$. Consequently, it can be seen as a stochastic form of the Newton update, which minimizes a random quadratic approximation at each iteration.

In this paper, we also analyze a *partially sketched Newton update*, which takes the following form. Given an additive decomposition of the form $f = f_0 + g$, we perform a sketch of of the Hessian $\nabla^2 f_0$ while retaining the exact form of the Hessian $\nabla^2 g$. This splitting leads to the partially sketched update

$$x^{t+1} := \arg\min_{x\in\mathcal{C}} \Big\{ \frac{1}{2}(x-x^t)^T Q^t (x-x^t) + \langle \nabla f(x^t),\, x-x^t\rangle \Big\}, \tag{3.4}$$

where $Q^t := (S^t \nabla^2 f_0(x^t)^{1/2})^T S^t \nabla^2 f_0(x^t)^{1/2} + \nabla^2 g(x^t)$.

For either the fully sketched (3.2) or partially sketched updates (3.4), our analysis shows that there are many settings in which the sketch dimension m can be chosen to be substantially smaller than n, in which cases the sketched Newton updates will be much cheaper than a standard Newton update. For instance, the unconstrained update (3.3) can be computed in at most $\mathcal{O}(md^2)$ time, as opposed to the $\mathcal{O}(nd^2)$ time of the standard Newton update. In constrained settings, we show that the sketch dimension m can often be chosen even smaller—even $m \ll d$—which leads to further savings.

3.2. Affine invariance of the Newton sketch and sketched KKT systems. A desirable feature of the Newton sketch is that, similar to the original Newton's method, both of its forms remain (statistically) invariant under an affine transformation. In other words, if we apply Newton sketch on an affine transformation of a particular function, the statistics of the iterates are related by the same transformation. As a concrete example, consider the problem of minimizing a function $f : \mathbb{R}^d \to \mathbb{R}$ subject to equality constraints $Cx = d$, for some matrix $C \in \mathbb{R}^{n \times d}$ and vector $d \in \mathbb{R}^n$. For this particular problem, the Newton sketch update takes the form

$$x^{t+1} := \arg \min_{Cx=d} \Big\{ \frac{1}{2} \|S^t \nabla^2 f(x^t)^{1/2}(x - x^t)\|_2^2 + \langle \nabla f(x^t),\, x - x^t \rangle \Big\}. \tag{3.5}$$

Equivalently, by introducing Lagrangian dual variables for the linear constraints, it is equivalent to solve the following *sketched KKT system*

$$\begin{bmatrix} (\nabla^2 f(x^t)^{1/2})^T (S^t)^T S^t \nabla^2 f(x^t)^{1/2} & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\text{NSK}} \\ w_{\text{NSK}} \end{bmatrix} = - \begin{bmatrix} \nabla f(x^t) \\ 0 \end{bmatrix}$$

where $\Delta x_{\text{NSK}} = x^{t+1} - x^t \in \mathbb{R}^d$ is the sketched Newton step where x^t is assumed feasible, and $w_{\text{NSK}} \in \mathbb{R}^n$ is the optimal dual variable for the stochastic quadratic approximation.

Now fix the random sketching matrix S^t and consider the transformed objective function $\widehat{f}(y) := f(By)$, where $B \in \mathbb{R}^{d \times d}$ is an invertible matrix. If we apply the Newton sketch algorithm to the transformed problem involving $\widehat{f}$, the sketched Newton step Δy_{NSK} is given by the solution to the system

$$\begin{bmatrix} B^T (\nabla^2 f(x^t)^{1/2})^T (S^t)^T S^t \nabla^2 f(x^t)^{1/2} B & B^T C^T \\ CB & 0 \end{bmatrix} \begin{bmatrix} \Delta y_{\text{NSK}} \\ \widehat{w}_{\text{NSK}} \end{bmatrix} = - \begin{bmatrix} B^T \nabla f(x^t) \\ 0 \end{bmatrix},$$

which shows that $B \Delta y_{\text{NSK}} = \Delta x_{\text{NSK}}$. Note that the upper-left block in the above matrix is has rank at most m, and consequently the above 2×2 block matrix has rank at most $m + \operatorname{rank}(C)$.

3.3. Some examples. In order to provide some intuition, let us provide some simple examples to which the sketched Newton updates can be applied.

EXAMPLE 1 (*Newton sketch for LP solving*). Consider a linear program (LP) in the standard form

$$\min_{Ax \leq b} \langle c,\, x \rangle \tag{3.6}$$

where $A \in \mathbb{R}^{n \times d}$ is a given constraint matrix. We assume that the polytope $\{x \in \mathbb{R}^d \mid Ax \leq b\}$ is bounded so that the minimum achieved. A barrier method approach to this LP is based on solving a sequence of problems of the form

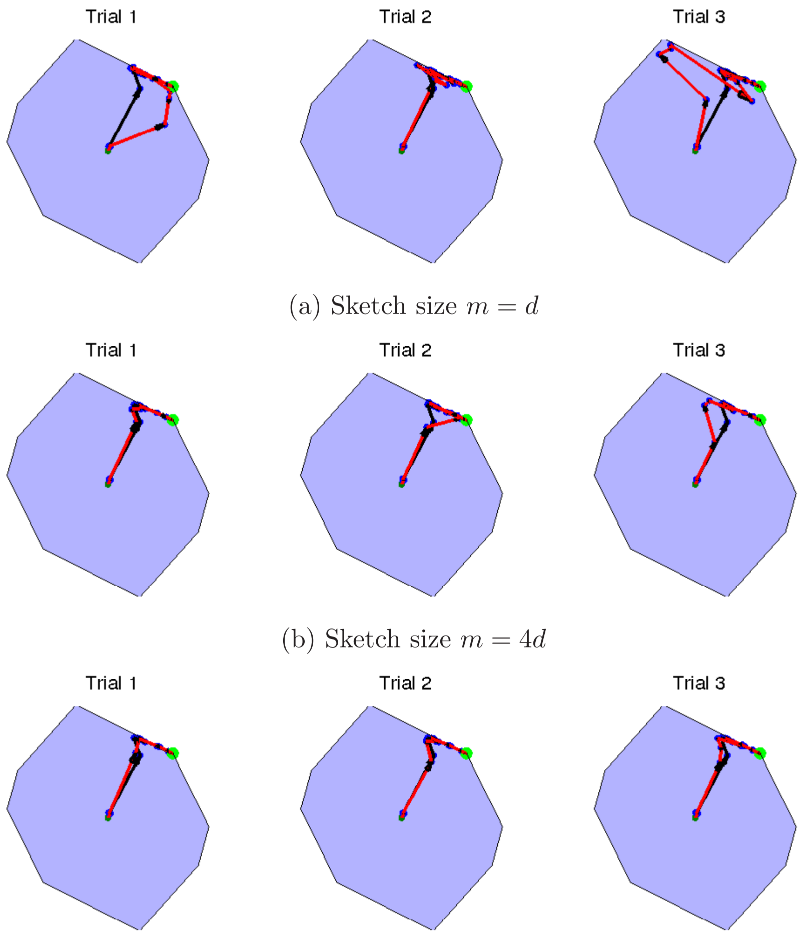
$$\min_{x \in \mathbb{R}^d} \Big\{ \underbrace{\tau \langle c,\, x \rangle - \sum_{i=1}^n \log(b_i - \langle a_i,\, x \rangle)}_{f(x)} \Big\},$$

where $a_i \in \mathbb{R}^d$ denotes the i^{th} row of A, and $\tau > 0$ is a weight parameter that is adjusted during the algorithm. By inspection, the function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is twice-differentiable, and its Hessian is given by $\nabla^2 f(x) = A^T \operatorname{diag}\big\{\frac{1}{(b_i - \langle a_i,\, x \rangle)^2}\big\} A$. A Hessian square root is given by $\nabla^2 f(x)^{1/2} := \operatorname{diag}\Big(\frac{1}{|b_i - \langle a_i,\, x \rangle|}\Big) A$, which allows us to compute the sketched version

$$S \nabla^2 f(x)^{1/2} = S \ \operatorname{diag}\left(\frac{1}{|b_i - \langle a_i,\, x \rangle|}\right) A.$$

With a ROS sketch matrix, computing this matrix requires $\mathcal{O}(nd\log(m))$ basic operations. The complexity of each Newton sketch iteration scales as $\mathcal{O}(md^2)$, where m is at most $\mathcal{O}(d)$. In contrast, the standard unsketched form of the Newton update has complexity $\mathcal{O}(nd^2)$, so that the sketched method is computationally cheaper whenever there are many more constraints than dimensions ($n > d$).

By increasing the barrier parameter τ, we obtain a sequence of solutions that approach the optimum to the LP, which we refer to as the central path. As a simple illustration, Figure 1 compares the central paths generated by the ordinary and sketched Newton updates for a polytope defined by $n = 32$ constraints in dimension $d = 2$. Each row shows three independent trials of the method for a given sketch dimension m; the top, middle and bottom rows correspond to sketch dimensions $m \in \{d, 4d, 16d\}$ respectively. Note that as the sketch dimension m is increased, the central path taken by the sketched updates converges to the standard central path.



(a) Sketch size $m = d$

(b) Sketch size $m = 4d$

(c) Sketch size $m = 16d$

Fig. 1. Comparisons of central paths for a simple linear program in two dimensions. Each row shows three independent trials for a given sketch dimension: across the rows, the sketch dimension ranges as $m \in \{d, 4d, 16d\}$. The black arrows show Newton steps taken by the standard interior point method, whereas red arrows show the steps taken by the sketched version. The green point at the vertex represents the optimum. In all cases, the sketched algorithm converges to the optimum, and as the sketch dimension m increases, the sketched central path converges to the standard central path.

As a second example, we consider the problem of maximum likelihood estimation for generalized linear models.

Example 2 (*Newton sketch for maximum likelihood estimation*). The class of generalized linear models (GLMs) is used to model a wide variety of prediction and classification problems, in which the goal is to predict some output variable $y \in \mathcal{Y}$ on the basis of a covariate vector $a \in \mathbb{R}^d$. GLMs include standard linear Gaussian model (in which $\mathcal{Y} = \mathbb{R}$), as well as logistic models for classification (in which

$\mathcal{Y} = \{-1, +1\}$), as well as as Poisson models for count-valued responses (in which $\mathcal{Y} = \{0, 1, 2, \ldots\}$) as special cases the. See the book [24] for further details and applications.

Given a collection of n observations $\{(y_i, a_i)\}_{i=1}^n$ of response-covariate pairs from some GLM, the problem of constrained maximum likelihood estimation be written in the form

$$\min_{x \in \mathcal{C}} \Big\{ \underbrace{\sum_{i=1}^n \psi(\langle a_i,\, x\rangle, y_i)}_{f(x)} \Big\}, \tag{3.7}$$

where $\psi : \mathbb{R} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a given convex function, and $\mathcal{C} \subset \mathbb{R}^d$ is a convex constraint set, chosen by the user to enforce a certain type of structure in the solution. Important special cases of GLMs include the linear Gaussian model, in which $\psi(u, y) = \frac{1}{2}(y - u)^2$, and the problem (3.7) corresponds to the method of least-squares, as well as the problem of logistic regression, obtained by setting $\psi(u, y) = \log(1 + \exp(-yu))$.

Letting $A \in \mathbb{R}^{n \times d}$ denote the data matrix with $a_i \in \mathbb{R}^d$ as its i^{th} row, the Hessian of the objective (3.7) takes the form

$$\nabla^2 f(x) = A^T \operatorname{diag}\left(\psi''(a_i^T x)\right)_{i=1}^n A$$

Since the function ψ is convex, we are guaranteed that $\psi''(a_i^T x) \geq 0$, and hence the $n \times d$ matrix $\operatorname{diag}\left(\psi''(a_i^T x)\right)^{1/2} A$ can be used as a matrix square-root. We return to explore this class of examples in more depth in Section 5.1.

3.4. Local convergence analysis using strong convexity. Returning now to the general setting, we begin by proving a local convergence guarantee for the sketched Newton updates. In particular, this theorem provides insight into how large the sketch dimension m must be in order to guarantee good local behavior of the sketched Newton algorithm.

Our analysis involves the geometry of the tangent cone of the optimal vector x^*. More precisely, given a constraint set $\mathcal{C}$ and the minimizer $x^* := \arg\min_{x \in \mathcal{C}} f(x)$, the tangent cone at x^* is given by

$$\mathcal{K} := \big\{\Delta \in \mathbb{R}^d \mid x^* + t\Delta \in \mathcal{C} \quad \text{for some } t > 0\big\}. \tag{3.8}$$

The local analysis to be given in this section involves the *cone-constrained eigenvalues of the Hessian* $\nabla^2 f(x^*)$, defined as

$$\gamma = \inf_{z \in \mathcal{K} \cap \mathcal{S}^{d-1}} \langle z,\, \nabla^2 f(x^*)) z\rangle, \quad \text{and} \quad \beta = \sup_{z \in \mathcal{K} \cap \mathcal{S}^{d-1}} \langle z,\, \nabla^2 f(x^*)) z\rangle. \tag{3.9}$$

In the unconstrained case ($\mathcal{C} = \mathbb{R}^d$), we have $\mathcal{K} = \mathbb{R}^d$, and so that γ and β reduce to the minimum and maximum eigenvalues of the Hessian $\nabla^2 f(x^*)$. In the classical analysis of Newton's method, these quantities measure the strong convexity and smoothness parameters of the function f. Note that the condition $\gamma > 0$ much weaker than strong convexity as it can hold for Hessian matrices that are rank-deficient, as long as the tangent cone $\mathcal{K}$ is suitably small.

Recalling the definition of the Gaussian width from Section 2.4, our choice of the sketch dimension m depends on the width of the renormalized tangent cone. In particular, for the following theorem, we require it to be lower bounded as

$$m \geq \frac{c}{\epsilon^2} \max_{x \in \mathcal{C}} \mathcal{W}^2(\nabla^2 f(x)^{1/2} \mathcal{K}), \tag{3.10}$$

where $\epsilon \in (0, \frac{\gamma}{9\beta})$ is a user-defined tolerance, and c is a universal constant. Since the Hessian square-root $\nabla^2 f(x)^{1/2}$ has dimensions $n \times d$, this squared Gaussian width is at at most $\min\{n, d\}$. This worst-case bound is achieved for an unconstrained problem (in which case $\mathcal{K} = \mathbb{R}^d$), but the Gaussian width can be substantially smaller for constrained problems. For instance, consider an equality constrained problem with affine constraint $Cx = b$. For such a problem, the tangent cone lies within the nullspace of the matrix C—say it is d_C-dimensional. It then follows that the squared Gaussian width (3.10) is also bounded by d_C; see the example following Theorem 3.1 for a concrete illustration. Other examples in which the Gaussian width can be substantially smaller include problems involving simplex constraints (portfolio optimization), or ℓ_1-constraints (sparse regression).

With this set-up, the following theorem is applicable to any twice-differentiable objective f with cone-constrained eigenvalues (γ, β) defined in equation (3.9), and with Hessian that is L-Lipschitz continuous, as defined in equation (2.2).

Theorem 3.1 (Local convergence of Newton Sketch). *For a given tolerance $\epsilon \in (0, \frac{2\gamma}{9\beta})$, consider the Newton sketch updates* (3.2) *based on an initialization x^0 such that $\|x^0 - x^*\|_2 \leq \frac{\gamma}{8L}$, and a sketch dimension m satisfying the lower bound* (3.10). *Then with probability at least $1 - c_1 N e^{-c_2 m}$, the Euclidean error satisfies the bound*

$$\|x^{t+1} - x^*\|_2 \leq \epsilon \frac{\beta}{\gamma} \|x^t - x^*\|_2 + \frac{4L}{\gamma} \|x^t - x^*\|_2^2, \qquad \textit{for iterations } t = 0, \ldots, N-1. \tag{3.11}$$

The bound (3.11) shows that when ϵ is small enough—say $\epsilon = \beta/4\gamma$—then the optimization error $\Delta^t = x^t - x^*$ decays at a linear-quadratic convergence rate. More specifically, the rate is initially quadratic—that is, $\|\Delta^{t+1}\|_2 \approx \frac{4L}{\gamma}\|\Delta^t\|_2^2$ when $\|\Delta^t\|_2$ is large. However, as the iterations progress and $\|\Delta^t\|_2$ becomes substantially less than 1, then the rate becomes linear—meaning that $\|\Delta^{t+1}\|_2 \approx \epsilon\frac{\beta}{\gamma}\|\Delta^t\|_2$—since the term $\frac{4L}{\gamma}\|\Delta^t\|_2^2$ becomes negligible compared to $\epsilon\frac{\beta}{\gamma}\|\Delta^t\|_2$. Unwrapping the recursion for all N steps, the linear rate guarantees the conservative error bounds

$$\|x^N - x^*\|_2 \leq \frac{\gamma}{8L}\Big(\frac{1}{2} + \epsilon\frac{\beta}{\gamma}\Big)^N, \quad \text{and} \quad f(x^N) - f(x^*) \leq \frac{\beta\gamma}{8L}\Big(\frac{1}{2} + \epsilon\frac{\beta}{\gamma}\Big)^N. \tag{3.12}$$

A notable feature of Theorem 3.1 is that, depending on the structure of the problem, the linear-quadratic convergence can be obtained using a sketch dimension m that is substantially smaller than $\min\{n, d\}$. As an illustrative example, we performed simulations for some instantiations of a portfolio optimization problem: it is a linearly-constrained quadratic program of the form

$$\min_{\substack{x \geq 0 \\ \sum_{j=1}^d x_j = 1}} \Big\{\frac{1}{2} x^T A^T A x - \langle c,\, x\rangle\Big\}, \tag{3.13}$$

where $A \in \mathbb{R}^{n \times d}$ and $c \in \mathbb{R}^d$ are matrices and vectors that arise from data (see Section 5.3 for more details). We used the Newton sketch to solve different sizes of this problem $d \in \{10, 20, 30, 40, 50, 60\}$, and with $n = d^3$ in each case. Each problem was constructed so that the optimal vector $x^* \in \mathbb{R}^d$ had at most $s = \lceil 2\log(d)\rceil$ non-zero entries. A calculation of the Gaussian width for this problem (see Appendix A for the details) shows that it suffices to take a sketch dimension $m \gtrsim s \log d$, and we implemented the algorithm with this choice. Figure 2 shows the convergence rate of the Newton sketch algorithm for the six different problem sizes: consistent with our theory, the sketch dimension $m \ll \min\{d, n\}$ suffices to guarantee linear convergence in all cases.

It is also possible obtain an asymptotically super-linear rate by using an iteration-dependent sketching accuracy $\epsilon = \epsilon(t)$. The following corollary summarizes one such possible guarantee:

Corollary 3.2. *Consider the Newton sketch iterates using the iteration-dependent sketching accuracy $\epsilon(t) = \frac{1}{\log(1+t)}$. Then with the same probability as in Theorem 3.1, we have*

$$\|x^{t+1} - x^*\|_2 \leq \frac{1}{\log(1+t)}\frac{\beta}{\gamma}\|x^t - x^*\|_2 + \frac{4L}{\gamma}\|x^t - x^*\|_2^2,$$

and consequently, super-linear convergence is obtained—namely, $\lim_{t\to\infty} \frac{\|x^{t+1} - x^*\|_2}{\|x^t - x^*\|_2} = 0$.

Note that the price for this super-linear convergence is that the sketch size is inflated by the factor $\epsilon^{-2}(t) = \log^2(1+t)$, so it is only logarithmic in the iteration number.

4. Newton sketch for self-concordant functions. The analysis and complexity estimates given in the previous section involve the curvature constants (γ, β) and the Lipschitz constant L, which are seldom known in practice. Moreover, as with the analysis of classical Newton method, the

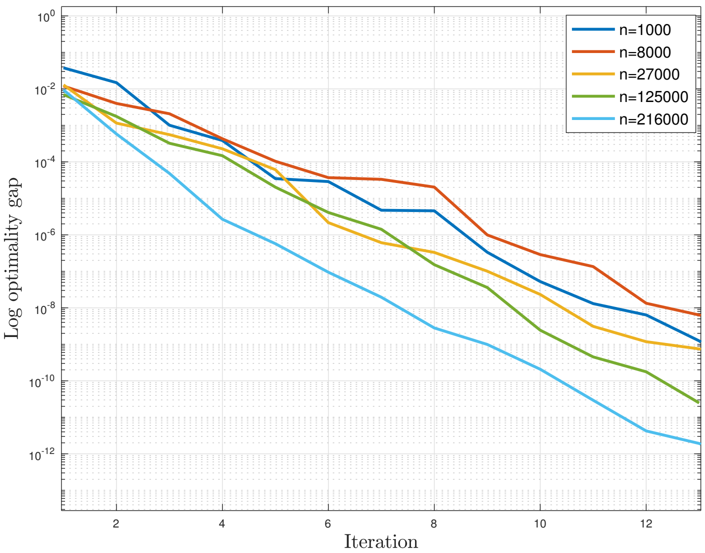


Fig. 2. Empirical illustration of the linear convergence of the Newton sketch algorithm for an ensemble of portfolio optimization problems (3.13). In all cases, the algorithm was implemented using a sketch dimension $m = \lceil 4s \log d \rceil$, where s is an upper bound on the number of non-zeros in the optimal solution x^*; this quantity satisfies the required lower bound (3.10), and consistent with the theory, the algorithm exhibits linear convergence.

theory is local, in that the linear-quadratic convergence takes place once the iterates enter a suitable basin of the origin.

In this section, we seek to obtain global convergence results that do not depend on unknown problem parameters. As in the classical analysis, the appropriate setting in which to seek such results is for self-concordant functions, and using an appropriate form of backtracking line search. We begin by analyzing the unconstrained case, and then discuss extensions to constrained problems with self-concordant barriers. In each case, we show that given a suitable lower bound on the sketch dimension, the sketched Newton updates can be equipped with global convergence guarantees that hold with exponentially high probability. Moreover, the total number of iterations does not depend on any unknown constants such as strong convexity and Lipschitz parameters.

4.1. Unconstrained case. In this section, we consider the unconstrained optimization problem $\min_{x \in \mathbb{R}^d} f(x)$, where f is a closed convex self-concordant function that is bounded below. A closed convex function $\phi : \mathbb{R} \to \mathbb{R}$ is said to be *self-concordant* if

$$|\phi'''(x)| \leq 2 \left(\phi''(x)\right)^{3/2}. \tag{4.1}$$

This definition can be extended to a function $f : \mathbb{R}^d \to \mathbb{R}$ by imposing this requirement on the univariate functions $\phi_{x,y}(t) := f(x + ty)$, for all choices of x, y in the domain of f. Examples of self-concordant functions include linear and quadratic functions and the negative logarithm. Moreover, the property of self-concordance is preserved under addition and affine transformations.

Our main result provides a bound on the total number of Newton sketch iterations required to obtain a δ-approximate solution without imposing any sort of initialization condition, as was done in our previous analysis. This bound scales proportionally to $\log(1/\delta)$ and inversely in a parameter ν that depends on sketching accuracy $\epsilon \in (0, \frac{1}{4})$ and backtracking parameters (a, b) via

$$\nu = ab \frac{\eta^2}{1 + (\frac{1+\epsilon}{1-\epsilon})\eta} \quad \text{where} \quad \eta = \frac{1}{8} \frac{1 - \frac{1}{2}(\frac{1+\epsilon}{1-\epsilon})^2 - a}{(\frac{1+\epsilon}{1-\epsilon})^3}. \tag{4.2}$$

With this set-up, we have the following guarantee:

Algorithm 1 Newton Sketch with backtracking line search

```
Input: Starting point $x^0$, tolerance $\delta > 0$, $(a, b)$ line-search parameters, sketching matrices $\{S^t\}_{t=0}^{\infty} \in \mathbb{R}^{m \times n}$.
  1: Compute approximate Newton step $\Delta x^t$ and approximate Newton decrement $\lambda(x)$
         $\Delta x^t := \arg \min_{\Delta + x^t \in \mathcal{C}} \langle \nabla f(x^t), \Delta \rangle + \frac{1}{2} \| S^t (\nabla^2 f(x^t))^{1/2} \Delta \|_2^2;$
         $\widetilde{\lambda}_f(x^t) := \nabla f(x)^T \Delta x^t.$
  2: Quit if $\tilde{\lambda}(x^t)^2/2 \leq \delta$.
  3: Line search: choose $\mu$ :    while $f(x^t + \mu \Delta x^t) > f(x^t) + a\mu\lambda(x^t)$, or $x^t + \mu\Delta \notin \mathcal{C}$    $\mu \leftarrow b\mu$
  4: Update: $x^{t+1} = x^t + \mu \Delta x^t$
Output: minimizer $x^t$, optimality gap $\lambda(x^t)$
```

Theorem 4.1. *Let f be a strictly convex self-concordant function and $\mathcal{C} = \mathbb{R}^d$. Given a sketching matrix $S \in \mathbb{R}^{m \times n}$ with $m = \frac{c_3}{\epsilon^2} \max_{x \in \mathrm{dom}\, f} \mathrm{rank}(\nabla^2 f(x))$, the number of total iterations T for obtaining a δ-approximate solution in function value via Algorithm 1 is at most*

$$N = \frac{f(x^0) - f(x^*)}{\nu} + 0.65 \log_2(\frac{1}{16\delta}), \tag{4.3}$$

with probability at least $1 - c_1 N e^{-c_2 m}$.

The iteration bound (4.3) shows that the convergence of the Newton sketch is independent of the properties of the function f and problem parameters, similar to classical Newton's method. Note that for problems with $n > d$, the complexity of each Newton sketch step is at most $\mathcal{O}(d^3 + nd \log d)$, which is smaller than that of Newton's Method ($\mathcal{O}(nd^2)$), and also smaller than typical first-order optimization methods ($\mathcal{O}(\kappa nd)$ per iteration) that depend on data conditioning whenever $n > d^2$, ignoring logarithmic factors.

4.1.1. Rank-deficient Hessians. As stated, Theorem 4.1 requires the function to be strictly convex. However, by exploiting the affine invariance of the Newton sketch updates, we can also obtain guarantees of the form (4.3) for the Newton sketch applied to problems with singular Hessians. As a concrete example, given a matrix $A \in \mathbb{R}^{n \times d}$ that is rank-deficient, i.e., with $\mathrm{rank}(A) = r < \min\{n, d\}$, consider a function of the form $f(x) = g(Ax)$, where $g : \mathbb{R}^n \to \mathbb{R}$ is strictly convex and self-concordant. Due to the rank-deficiency of A, the Hessian of f will also be rank-deficient, so that Theorem 4.1 does not directly apply. However, suppose that we let $A = U \Sigma V^T$ be the full singular value decomposition of A, where Σ is a diagonal matrix with $\Sigma_{jj} = 0$ for all indices $j > r$. With this notation, define the function $\widehat{f}(y) = g(AVy)$, corresponding to the invertible transformation $x = Vy$. Note that as a result of the affine invariance property, we don't have to compute the SVD explicitly and perform this transformation. We then have

$$\widehat{f}(y) = g(U\Sigma y) \;=\; g(U\Sigma_{1:r} y_{1:r}),$$

where $y_{1:r} \in \mathbb{R}^r$ denotes the subvector of the first r entries of y. Hence, viewed as a function on $\mathbb{R}^r$, the transformed function $\widehat{f}$ is strictly convex and self-concordant, so that Theorem 4.1 can be applied. By the affine invariance property, the Newton sketch applied to the original function f has the same convergence guarantees (and transformed iterates) as the reduced strictly convex function. Consequently, the sketch size choice $m = \frac{c}{\epsilon^2} \mathrm{rank}(A)$ is sufficient. Note that in many applications, the rank of A can be much smaller than $\min(n, d)$, and so that the Newton sketch complexity $\mathcal{O}(m^2 d)$ is correspondingly smaller, relative to other schemes that do not exploit the low-rank structure. Some optimization methods can exploit low-rankness when a factorization of the form $A = LR$ is available. However, note that the cost of computing such a low rank factorization scales as $\mathcal{O}(nd^2)$, which dominates the overall complexity of Newton sketch, including sketching time.

4.2. Newton sketch with self-concordant barriers. We now turn to the more general constrained case. Given a closed, convex self-concordant function $f_0 : \mathbb{R}^d \to \mathbb{R}$, let $\mathcal{C}$ be a convex subset of $\mathbb{R}^d$, and consider the constrained optimization problem $\min_{x \in \mathcal{C}} f_0(x)$. If we are given a convex self-concordant barrier function $g(x)$ for the constraint set $\mathcal{C}$, it is customary to consider the unconstrained

Algorithm 2 Newton Sketch with self-concordant barriers

```
Input: Starting point $x^0$, constraint $\mathcal{C}$, corresponding barrier function $g$ such that $f = f_0 + g$, tolerance $\delta > 0$, $(\alpha, \beta)$ line-search
    parameters, sketching matrices $S^t \in \mathbb{R}^{m \times n}$.
 1: Compute approximate Newton step $\Delta x^t$ and approximate Newton decrement $\widetilde{\lambda}_f$.

        $\Delta x^t := \arg \min_{x^t + \Delta \in \mathcal{C}} \; \langle \nabla f(x^t), \, \Delta \rangle + \frac{1}{2} \| S^t (\nabla^2 f_0(x^t))^{1/2} \Delta \|_2^2 + \frac{1}{2} \Delta^T \nabla^2 g(x^t) \Delta;$
        $\widetilde{\lambda}_f(x^t) := \nabla f(x)^T \Delta x^t$

 2: Quit if $\tilde{\lambda}(x^t)^2/2 \leq \delta$.
 3: Line search: choose $\mu$ :    while $f(x^t + \mu \Delta x^t) > f(x^t) + \alpha \mu \lambda(x^t)$, or $x^t + \mu \Delta \notin \mathcal{C}$    $\mu \leftarrow \beta \mu$.
 4: Update: $x^{t+1} = x^t + \mu \Delta x^t$.
Output: minimizer $x^t$, optimality gap $\lambda(x^t)$.
```

and penalized problem

$$\min_{x \in \mathbb{R}^d} \Big\{ \underbrace{f_0(x) + g(x)}_{f(x)} \Big\},$$

which approximates the original problem. One way in which to solve this unconstrained problem is by sketching the Hessian of both f_0 and g, in which case the theory of the previous section is applicable. However, there are many cases in which the constraints describing $\mathcal{C}$ are relatively simple, and so the Hessian of g is highly-structured. For instance, if the constraint set is the usual simplex (i.e., $x \geq 0$ and $\langle 1, \, x \rangle \leq 1$), then the Hessian of the associated log barrier function is a diagonal matrix plus a rank one matrix. Other examples include problems for which g has a separable structure; such functions frequently arise as regularizers for ill-posed inverse problems. Examples of such regularizers include ℓ_2 regularization $g(x) = \frac{1}{2}\|x\|_2^2$, graph regularization $g(x) = \frac{1}{2} \sum_{i,j \in E} (x_i - x_j)^2$ induced by an edge set E (e.g., finite differences) and also other differentiable norms $g(x) = \Big(\sum_{i=1}^d x_i^p \Big)^{1/p}$ for $1 < p < \infty$.

In all such cases, an attractive strategy is to apply a *partial Newton sketch*, in which we sketch the Hessian term $\nabla^2 f_0(x)$ and retain the exact Hessian $\nabla^2 g(x)$, as in the previously described updates (3.4). More formally, Algorithm 2 provides a summary of the steps, including the choice of the line search parameters. The main result of this section provides a guarantee on this algorithm, assuming that the sequence of sketch dimensions $\{m^t\}_{t=0}^{\infty}$ is appropriately chosen.

The choice of sketch dimensions depends on the tangent cones defined by the iterates, namely the sets

$$\mathcal{K}^t := \big\{ \Delta \in \mathbb{R}^d \mid x^t + \alpha \Delta \in \mathcal{C} \quad \text{for some } \alpha > 0 \big\}.$$

For a given sketch accuracy $\epsilon \in (0, 1)$, we require that the sequence of sketch dimensions satisfies the lower bound

$$m^t \geq \frac{c_3}{\epsilon^2} \max_{x \in \mathcal{C}} \mathcal{W}^2 (\nabla^2 f(x)^{1/2} \mathcal{K}^t). \tag{4.4}$$

Finally, the reader should recall the parameter ν was defined in equation (4.2), which depends only on the sketching accuracy ϵ and the line search parameters. Given this set-up, we have the following guarantee:

THEOREM 4.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex and self-concordant function, and let $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ be a convex and self-concordant barrier for the convex set $\mathcal{C}$. Suppose that we implement Algorithm 2 with sketch dimensions $\{m^t\}_{t \geq 0}$ satisfying the lower bound (4.4). Then performing*

$$N = \frac{f(x^0) - f(x^*)}{\nu} + 0.65 \log_2 \Big(\frac{1}{16\delta} \Big) \qquad \textit{iterations}$$

suffices to obtain δ-approximate solution in function value with probability at least $1 - c_1 N e^{-c_2 m}$.

Thus, we see that the Newton Sketch method can also be used with self-concordant barrier functions, which considerably extends its scope. In the above theorem, note that we can isolate affine constraints

Algorithm 3 Interior point methods using Newton Sketch

```
Input: Strictly feasible starting point $x^0$, initial parameter $\tau^0$ s.t. $\tau := \tau^0 > 0$, $\mu > 1$, tolerance $\delta > 0$.
 1: Centering step: Compute $\widehat{x}(\tau)$ by Newton Sketch with backtracking line-search initialized at $x$
    using Algorithm 1 or Algorithm 2.
 2: Update $x := \widehat{x}(\tau)$.
 3: Quit if $r/\tau \leq \delta$.
 4: Increase $\tau$ by $\tau := \mu\tau$.
Output: minimizer $\widehat{x}(\tau)$.
```

from $\mathcal{C}$ and enforce them at each Newton step. Section 5.6 provides a numerical illustration of its performance in this context. As we discuss in the next section, there is a flexibility in choosing the decomposition f_0 and g corresponding to objective and barrier, which enables us to also sketch the constraints.

4.3. Sketching with interior point methods. In this section, we discuss the application of Newton Sketch to a form of barrier or interior point methods. In particular we discuss two different strategies and provide rigorous worst-case complexity results when the functions in the objective and constraints are self-concordant. More precisely, let us consider a problem of the form

$$\min_{x \in \mathbb{R}^d} \; f_0(x) \quad \text{subject to} \quad g_j(x) \leq 0 \quad \text{for } j = 1, \ldots, r, \tag{4.5}$$

where f_0 and $\{g_j\}_{j=1}^r$ are twice-differentiable convex functions. We assume that there exists a unique solution x^* to the above problem.

The barrier method for computing x^* is based on solving a sequence of problems of the form

$$\widehat{x}(\tau) := \arg\min_{x \in \mathbb{R}^d} \Big\{ \tau f_0(x) - \sum_{j=1}^{r} \log(-g_j(x)) \Big\}, \tag{4.6}$$

for increasing values of the parameter $\tau \geq 1$. The family of solutions $\{\widehat{x}(\tau)\}_{\tau \geq 1}$ trace out what is known as the central path. A standard bound (e.g., [6]) on the sub-optimality of $\widehat{x}(\tau)$ is given by

$$f_0(\widehat{x}(\tau)) - f_0(x^*) \leq \frac{r}{\tau}.$$

The barrier method successively updates the penalty parameter τ and also the starting points supplied to Newton's method using previous solutions.

Since Newton's method lies at the heart of the barrier method, we can obtain a fast version by replacing the exact Newton minimization with the Newton sketch. Algorithm 3 provides a precise description of this strategy. As noted in Step 1, there are two different strategies in dealing with the convex constraints $g_j(x) \leq 0$ for $j = 1, \ldots, r$:

- *Full sketch:* Sketch the full Hessian of the objective function (4.6) using Algorithm 1 ,
- *Partial sketch:* Sketch only the Hessians corresponding to a subset of the functions $\{f_0, g_j, j = 1, \ldots, r\}$, and use exact Hessians for the other functions. Apply Algorithm 2.

As shown by our theory, either approach leads to the same convergence guarantees, but the associated computational complexity can vary depending both on how data enters the objective and constraints, as well as the Hessian structure arising from particular functions. The following theorem is an application of the classical results on the barrier method tailored for Newton Sketch using any of the above strategies (e.g., see Boyd and Vandenberghe [6]). As before, the key parameter ν was defined in Theorem 4.1.

Theorem 4.3 (Newton Sketch complexity for interior point methods). *For a given target accuracy $\delta \in (0, 1)$ and any $\mu > 1$, the total number of Newton Sketch iterations required to obtain a δ-approximate solution using Algorithm 3 is at most*

$$\left\lceil \frac{\log\left(r/(\tau^0\delta)\right)}{\log \mu} \right\rceil \left(\frac{r(\mu - 1 - \log \mu)}{\nu} + 0.65 \log_2\left(\frac{1}{16\delta}\right) \right). \tag{4.7}$$

If the parameter μ is set to minimize the above upper-bound, the choice $\mu = 1 + \frac{1}{r}$ yields $\mathcal{O}(\sqrt{r})$ iterations. However, this "optimal" choice is typically not used in practice when applying the standard

Newton method; instead, it is common to use a fixed value of $\mu \in [2, 100]$. In experiments, experience suggests that the number of Newton iterations needed is a constant independent of r and other parameters. Theorem 4.3 allows us to obtain faster interior point solvers with rigorous worst-case complexity results. We show different applications of Algorithm 3 in the following section.

5. Applications and numerical results.

In this section, we discuss some applications of the Newton sketch to different optimization problems. In particular, we show various forms of Hessian structure that arise in applications, and how the Newton sketch can be computed. When the objective and/or the constraints contain more than one term, the barrier method with Newton Sketch has some flexibility in sketching. We discuss the choices of partial Hessian sketching strategy in the barrier method. It is also possible to apply the sketch in the primal or dual form, and we provide illustrations of both strategies here.

5.1. Estimation in generalized linear models.

Recall the problem of (constrained) maximum likelihood estimation for a generalized linear model, as previously introduced in Example 2. It leads to the family of optimization problems (3.7): here $\psi : \mathbb{R} \to \mathbb{R}$ is a given convex function arising from the probabilistic model, and $\mathcal{C} \subseteq \mathbb{R}^d$ is a closed convex set that is used to enforce a certain type of structure in the solution. Popular choices of such constraints include ℓ_1-balls (for enforcing sparsity in a vector), nuclear norms (for enforcing low-rank structure in a matrix), and other non-differentiable semi-norms based on total variation (e.g., $\sum_{j=1}^{d-1} |x_{j+1} - x_j|$), useful for enforcing smoothness or clustering constraints.

Suppose that we apply the Newton sketch algorithm to the optimization problem (3.7). Given the current iterate x^t, computing the next iterate x^{t+1} requires solving the constrained quadratic program

$$\min_{x \in \mathcal{C}} \left\{ \frac{1}{2} \|S \operatorname{diag}\left(\psi''(\langle a_i,\, x^t\rangle, y_i)\right)^{1/2} A(x - x^t)\|_2^2 + \sum_{i=1}^{n} \langle x,\, \psi'(\langle a_i,\, x^t\rangle, y_i)\rangle \right\}. \tag{5.1}$$

When the constraint $\mathcal{C}$ is a scaled version of the ℓ_1-ball—that is, $\mathcal{C} = \{x \in \mathbb{R}^d \mid \|x\|_1 \leq R\}$ for some radius $R > 0$—the convex program (5.1) is an instance of the Lasso program [36], for which there is a very large body of work. For small values of R, where the cardinality of the solution x is very small, an effective strategy is to apply a homotopy type algorithm, also known as LARS [15, 17], which solves the optimality conditions starting from $R = 0$. For other sets $\mathcal{C}$, another popular choice is projected gradient descent, which is efficient when projection onto $\mathcal{C}$ is computationally simple.

Focusing on the ℓ_1-constrained case, let us consider the problem of choosing a suitable sketch dimension m. Our choice involves the ℓ_1-restricted minimal eigenvalue of the data matrix $A^T A$, which is given by[†]

$$\gamma_s^-(A) := \min_{\substack{\|z\|_2 = 1 \\ \|z\|_1 \leq 2\sqrt{s}}} \|Az\|_2^2. \tag{5.2}$$

Note that we are always guaranteed that $\gamma_s^-(A) \geq \lambda_{\min}(A^T A)$. Our result also involves certain quantities that depend on the function ψ, namely

$$\psi''_{\min} := \min_{x \in \mathcal{C}} \min_{i=1,\ldots,n} \psi''(\langle a_i,\, x\rangle, y_i), \quad \text{and} \quad \psi''_{\max} := \max_{x \in \mathcal{C}} \max_{i=1,\ldots,n} \psi''(\langle a_i,\, x\rangle, y_i),$$

where $a_i \in \mathbb{R}^d$ is the i^{th} row of A. With this set-up, supposing that the optimal solution x^* has cardinality at most $\|x^*\|_0 \leq s$, then it can be shown (see Lemma A.1 in Appendix A) that it suffices to take a sketch size

$$m = c_0 \, \frac{\psi''_{\max}}{\psi''_{\min}} \frac{\max\limits_{j=1,\ldots,d} \|A_j\|_2^2}{\gamma_s^-(A)} \, s \log d, \tag{5.3}$$

where c_0 is a universal constant. Let us consider some examples to illustrate:

- Least-Squares regression: $\psi(u) = \frac{1}{2}u^2$, $\psi''(u) = 1$ and $\psi''_{\min} = \psi''_{\max} = 1$.

[†] Our choice of introducting the factor of two in the the constraint $\|z\|_1 \leq 2\sqrt{s}$ is for later theoretical convenience, due to the structure of the tangent cone associated with the ℓ_1-norm [31, 30].

- Poisson regression: $\psi(u) = e^u$, $\psi''(u) = e^u$ and $\frac{\psi''_{\max}}{\psi''_{\min}} = \frac{e^{RA_{\max}}}{e^{-RA_{\min}}}$
- Logistic regression: $\psi(u) = \log(1 + e^u)$, $\psi''(u) = \frac{e^u}{(e^u+1)^2}$ and $\frac{\psi''_{\max}}{\psi''_{\min}} = \frac{e^{RA_{\min}}}{e^{-RA_{\max}}} \frac{(e^{-RA_{\max}}+1)^2}{(e^{RA_{\min}}+1)^2}$,

where $A_{\max} := \max\limits_{i=1,\dots,n} \|a_i\|_\infty$, and $A_{\min} := \min\limits_{i=1,\dots,n} \|a_i\|_\infty$.

For a large class of distributions of data matrices, the sketch size choice given in equation (5.3) scales as $\mathcal{O}(s \log d)$. As an example, consider data matrices $A \in \mathbb{R}^{n \times d}$ where each row is independently sampled from a sub-Gaussian distribution with parameter one (see equation (2.3)). Then standard results on random matrices [37] show that $\gamma_s^-(A) > 1/2$ with high probability as long as $n > c_1 s \log d$ for a sufficiently large constant c_1. In addition, we have $\max\limits_{j=1,\dots,d} \|A_j\|_2^2 = \mathcal{O}(n)$, as well as $\frac{\psi''_{\max}}{\psi''_{\min}} = \mathcal{O}(\log(n))$. For such problems, the per iteration complexity of Newton Sketch update scales as $\mathcal{O}(s^2 d \log^2(d))$ using standard Lasso solvers (e.g., [20]) or as $\mathcal{O}(sd\log(d))$ using projected gradient descent, per gradient evaluation. Using ROS sketches and standard lasso interior point Lasso solvers to solve sketched Newton updates, the total complexity is therefore $(\mathcal{O}(s^2 d \log^2(d) + nd \log(d)) \log(1/\epsilon))$. This scaling can be substantially smaller than conventional algorithms that fail to exploit the small intrinsic dimension of the tangent cone.

5.2. Semidefinite programs. The Newton sketch can also be applied to semidefinite programs. As one illustration, let us consider a metric learning problem studied in machine learning. Suppose that we are given d-dimensional feature vectors $\{a_i\}_{i=1}^n$ and a collection of $\binom{n}{2}$ binary indicator variables $y_{ij} \in \{-1, +1\}^n$ given by

$$y_{ij} = \begin{cases} +1 & \text{if } a_i \text{ and } a_j \text{ belong to the same class} \\ -1 & \text{otherwise,} \end{cases}$$

defined for all distinct indices $i, j \in \{1, \dots, n\}$. The task is to estimate a positive semidefinite matrix X such that the semi-norm $\|(a_i - a_j)\|_X := \sqrt{\langle a_i - a_j,\, X(a_i - a_j)\rangle}$ is a good predictor of whether or not vectors i and j belong to the same class. Using the least-squares loss, one way in which to do so is by solving the semidefinite program (SDP)

$$\min_{X \succeq 0} \Big\{ \sum_{i \neq j}^{\binom{n}{2}} \big(\langle X,\, (a_i - a_j)(a_i - a_j)^T\rangle - y_{ij}\big)^2 + \lambda \operatorname{trace}(X) \Big\}.$$

Here the term trace(X), along with its multiplicative pre-factor $\lambda > 0$ that can be adjusted by the user, is a regularization term for encouraging a relatively low-rank solution. Using the standard self-concordant barrier $X \mapsto \log\det(X)$ for the PSD cone, the barrier method involves solving a sequence of sub-problems of the form

$$\min_{X \in \mathbb{R}^{d \times d}} \Big\{ \underbrace{\tau \sum_{i=1}^{n} (\langle X,\, a_i a_i^T\rangle - y_i)^2 + \tau\lambda \operatorname{trace} X - \log\det(X)}_{f(\operatorname{vec}(X))} \Big\}.$$

Now the Hessian of the function $\operatorname{vec}(X) \mapsto f(\operatorname{vec}(X))$ is a $d^2 \times d^2$ matrix given by

$$\nabla^2 f\big(\operatorname{vec}(X)\big) = \tau \sum_{i \neq j}^{\binom{n}{2}} \operatorname{vec}(A_{ij}) \operatorname{vec}(A_{ij})^T + X^{-1} \otimes X^{-1},$$

where $A_{ij} := (a_i - a_j)(a_i - a_j)^T$. Then we can apply the barrier method with partial Hessian sketch on the first term, $\{S_{ij}\operatorname{vec}(A_{ij})\}_{i \neq j}$ and exact Hessian for the second term. Since the vectorized decision variable is $\operatorname{vec}(X) \in \mathbb{R}^{d^2}$ the complexity of Newton Sketch is $\mathcal{O}(m^2 d^2)$ while the complexity of a classical SDP interior-point solver is $\mathcal{O}(nd^4)$ in practice.

5.3. Portfolio optimization and SVMs. Here we consider the Markowitz formulation of the portfolio optimization problem [22]. The objective is to find a vector $x \in \mathbb{R}^d$ belonging to the unit simplex, corresponding to non-negative weights associated with each of d possible assets, so as to maximize the expected return minus a coefficient times the variance of the return. Letting $\mu \in \mathbb{R}^d$ denote a vector corresponding to mean return of the assets, and we let $\Sigma \in \mathbb{R}^{d\times d}$ be a symmetric, positive semidefinite matrix which represents the covariance of the returns. The optimization problem is given by

$$\max_{x\geq 0,\, \sum_{j=1}^d x_j \leq 1} \Big\{ \mu^T x - \lambda \frac{1}{2}\, x^T \Sigma x \Big\}. \tag{5.4}$$

The covariance of returns is often estimated from past stock data via an empirical covariance matrix of the form $\Sigma = A^T A$; here columns of A are time series corresponding to assets normalized by $\sqrt{n}$, where n is the length of the observation window.

The barrier method can be used solve the above problem by solving penalized problems of the form

$$\min_{x\in\mathbb{R}^d} \Big\{ \underbrace{-\tau\, \mu^T x + \tau\lambda\, \frac{1}{2} x^T A^T A x - \sum_{i=1}^d \log(e_i^T x) - \log(1 - 1^T x)}_{f(x)} \Big\},$$

where $e_i \in \mathbb{R}^d$ is the i^{th} element of the canonical basis and 1 is a row vector of all-ones. Then the Hessian of the above barrier penalized formulation can be written as

$$\nabla^2 f(x) = \tau\lambda\, A^T A + \big(\operatorname{diag}\{x_i^2\}_{i=1}^d\big)^{-1} + 11^T.$$

Consequently, we can sketch the data dependent part of the Hessian via $\tau\lambda S A$ which has at most rank m and keep the remaining terms in the Hessian exact. Since the matrix 11^T is rank one, the resulting sketched estimate is therefore diagonal plus rank $(m+1)$ where the matrix inversion lemma [16] can be applied for efficient computation of the Newton Sketch update. Therefore, as long as $m \leq d$, the complexity per iteration scales as $\mathcal{O}(md^2)$, which is cheaper than the $\mathcal{O}(nd^2)$ per step complexity associated with classical interior point methods. We also note that support vector machine classification problems with squared hinge loss also has the same form as in equation (5.4), so that the same strategy can be applied.

5.4. Unconstrained logistic regression with $d \ll n$. Let us now turn to some numerical comparisons of the Newton Sketch with other popular optimization methods for large-scale instances of logistic regression. More specifically, we generated a data matrix $A \in \mathbb{R}^{n\times d}$ with $d = 100$ features and $n = 65536$ observations. Each row $a_i \in \mathbb{R}^d$ was generated from the d-variate Gaussian distribution $N(0, \Sigma)$ where the covariance matrix Σ has 1 on diagonals and ρ on off-diagonals. Consequently, we solve the optimization problem,

$$\min_{x\in\mathcal{C}} \sum_{i=1}^n \log(1 + \exp(a_i^T x y_i), \tag{5.5}$$

which is a special case of the GLM maximum likelihood problem given in (3.7) using Newton sketch (Algorithm 1) other optimization algorithms where $\mathcal{C} = \mathbb{R}^d$. As shown in Figure 3, the convergence of the algorithm per iteration is very similar to Newton's method. Besides the original Newton's method, the other algorithms compared are

- Gradient Descent (GD) with backtracking line search
- Stochastic Average Gradient (SAG) with line search
- Broyden-Fletcher-Goldfarb-Shanno algorithm (BFGS) (MATLAB R2015a implementation)
- Truncated Newton's Method (trunNewt)

We ran the Algorithm 1 with ROS sketch and sketch size $m = 4d$, line-search parameters $a = 0.1$ and $b = 0.5$ and plot iterates over 10 independent trials. The step size in the gradient method is computed using backtracking line search. For the Truncated Newton's Method, we first performed experiments by setting the maximum CG iteration number in the range $\{\log(d), 2\log(d), 3\log(d)..., 10\log(d)\}$, and then also implemented the residual stopping rule with accuracy $1/t$, where t is the iteration count, as suggested in [12]. The best choice among these parameters is shown as *trunNewt* in the plots. All algorithms are implemented in MATLAB (R2015a). In the plots, each iteration of the SAG algorithm corresponds to a pass over the data, which is of comparable complexity to a single iteration of GD. In order to keep the plots relatively uncluttered, we have excluded Stochastic Gradient Descent since it is dominated by another stochastic first-order method (SAG), and Accelerated Gradient Method [26] as it is quite similar to Gradient Descent. Plots on the left in Figure 3—that is panels (a), (c) and (e)—show the log duality gap versus the number of iterations: as expected, on this scale, the classical form of Newton's method is the fastest. However, when the log optimality gap is plotted versus the wall-clock time (right-side panels (b), (d) and (e)), we now see that the Newton sketch is the fastest. The panels (a) and (b) exhibit the case when there is no correlation ($\rho = 0$). For these very well-conditioned problems first-order methods are often the best. However, panels (c) and (d) exhibit the case when correlation is moderate ($\rho = 0.5$) where it can be seen that Newton sketch is the fastest method. Panels (e) and (f) further demonstrates that Newton sketch performs well even with high correlations ($\rho = 0.9$).

On the other hand, Figure 4 reveals the sensitivity of first order and stochastic gradient type methods to the distribution of the covariates. For these experiments, we generated a feature matrix A with $d = 100$ features and $n = 65536$ observations where each row $a_i \in \mathbb{R}^d$ was generated from the Student's t-distribution with covariance Σ. The covariance matrix Σ has 1 on the diagonal and ρ off the diagonal. The distribution of the data rows generated from Student's t-distribution is more heavy-tailed compared to a normal distribution. As it can be seen in Figure 3, SAG and GD perform quite poor compared to Figure 3 under the heavy-tailed distribution even in the uncorrelated case ($\rho = 0$). However, the performance of the Newton sketch is not changed by the distribution or the conditioning of the data and so outperforms other methods as predicted by our theory.

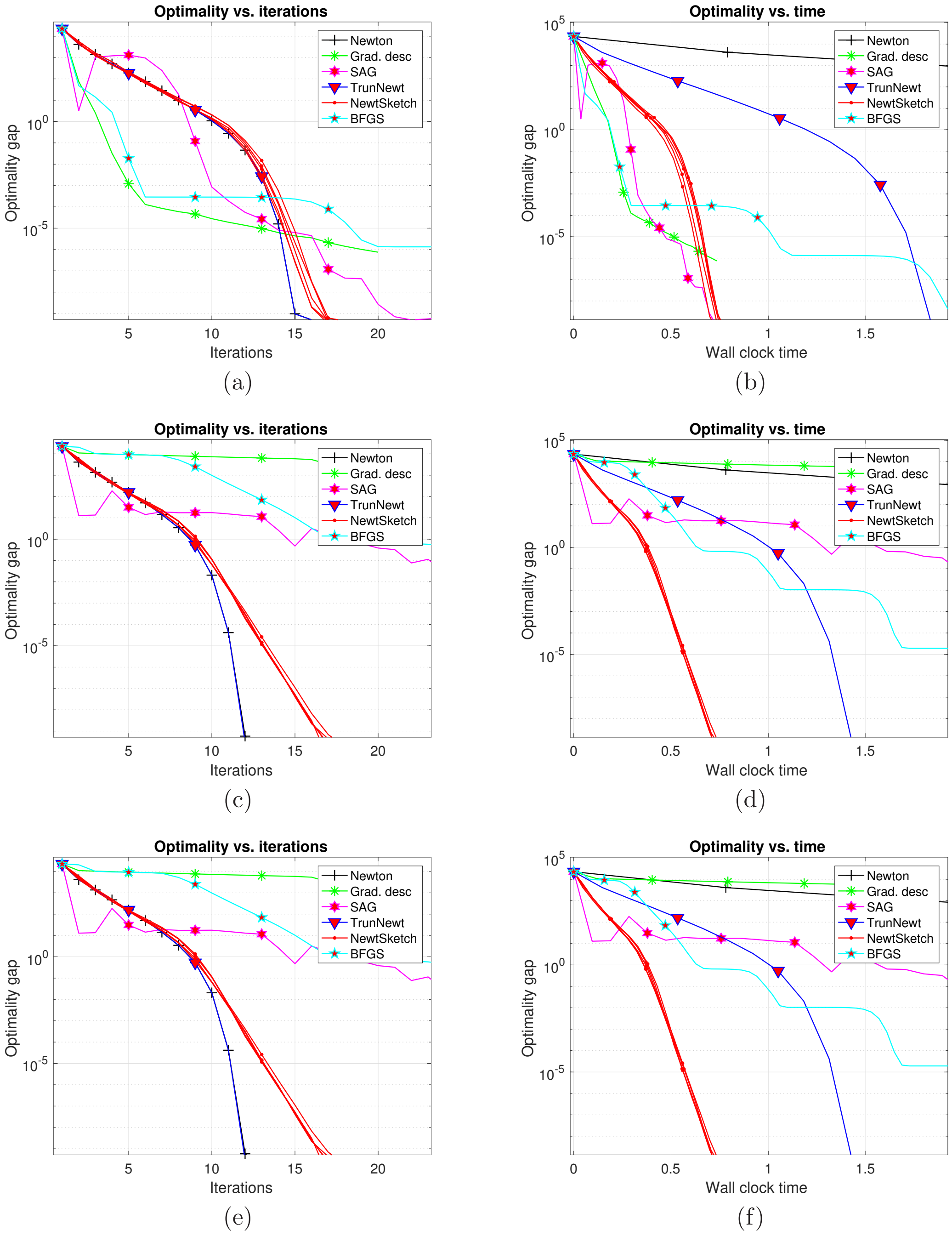
5.5. ℓ_1-constrained logistic regression and data conditioning. Next we provide some numerical comparisons of Newton Sketch, Newton's Method and Projected Gradient Descent when applied to an ℓ_1-constrained form of logistic regression. We consider the optimization problem (5.5) where the constraint set $\mathcal{C}$ is a scaled ℓ_1 ball. More specifically, we first generate a feature matrix $A \in \mathbb{R}^{n \times d}$ based on $d = 100$ features and $n = 1000$ observations. Each row $a_i \in \mathbb{R}^d$ is drawn from the d-variate Gaussian distribution $N(0, \Sigma)$; the covariance matrix has entries of the form $\Sigma_{ij} = 2|\rho|^{i-j}$, where $\rho \in [0, 1)$ is a parameter controlling the correlation, and hence the condition number of the data. For 10 different values of ρ we solved the ℓ_1-constrained problem ($\|x\|_1 \leq 0.1$), performing 200 independent trials (regenerating the data and sketching matrices randomly each time). The Newton and sketched Newton steps (Algorithm 1) are solved exactly using the homotopy algorithm—that is, the Lasso modification of the LARS updates [29, 15] using a Matlab implementation [34]. The homotopy method is very effective when the solution is very sparse. The ROS sketch with a sketch size of $m = \lceil 4 \times 10 \log d \rceil$ is used where 10 is the estimated cardinality of solution. As shown in Figure 5, Newton Sketch converges in about 6 (± 2) iterations independent of data conditioning while the exact Newton's method converges in 3 (± 1) iterations. However the number of iterations needed for projected gradient with line search increases steeply as ρ increases. Note that, ignoring logarithmic terms, the projected gradient and Newton Sketch have similar computational complexity ($\mathcal{O}(nd)$) per iteration while the Newton's method has higher computational complexity ($\mathcal{O}(nd^2)$).

5.6. A dual example: Lasso with $d \gg n$. The regularized Lasso problem takes the form $\min\limits_{x \in \mathbb{R}^d} \left\{ \frac{1}{2} \|Ax - y\|_2^2 + \lambda \|x\|_1 \right\}$, where $\lambda > 0$ is a user-specified regularization parameter. In this section, we consider efficient sketching strategies for this class of problems in the regime $d \gg n$. In particular, let us consider the corresponding dual program, given by

$$\max_{\|A^T w\|_\infty \leq \lambda} \left\{ -\frac{1}{2} \|y - w\|_2^2 \right\}.$$

By construction, the number of constraints d in the dual program is larger than the number of optimization variables n. If we apply the barrier method to solve this dual formulation, then we need to solve a sequence of problems of the form

$$\min_{w \in \mathbb{R}^n} \underbrace{\left\{ \tau \|y - w\|_2^2 - \sum_{j=1}^d \log(\lambda - \langle A_j, w \rangle) - \sum_{j=1}^d \log(\lambda + \langle A_j, w \rangle) \right\}}_{f(x)},$$



(a) (b)

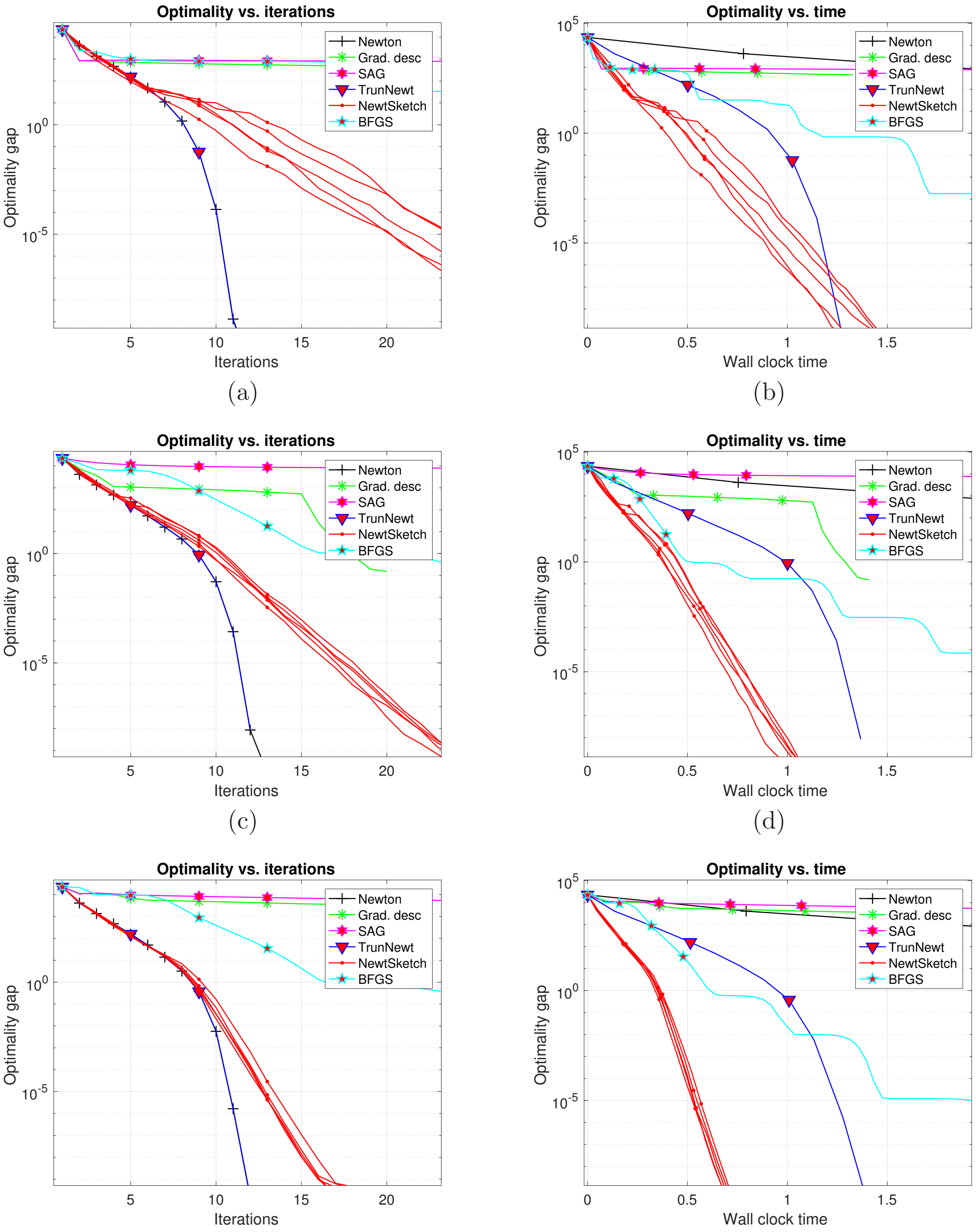
(c) (d)

(e) (f)

Fig. 3. Comparison of Newton sketch with various other algorithms, including Newton's Method, Truncated Newton, Stochastic Average Gradient and BFGS in the logistic regression problem with Gaussian data. Plots on the left show log optimality gap versus iteration number, and plots on the right show the log optimality gap versus wall-clock time (bottom). (a), (b): No correlation $\rho = 0$. For these very well-conditioned problems first-order methods (gradient descent and SAG) are often the best. (c), (d) Correlation $\rho = 0.7$. Newton sketch is now the best method. (e), (f) Correlation $\rho = 0.9$. Newton sketch performs well even with high correlations.

where $A_j \in \mathbb{R}^n$ denotes the j^{th} column of A. The Hessian of the above barrier penalized formulation can be written as

$$\nabla^2 f(w) = \tau I_n + A \,\mathrm{diag}\left(\frac{1}{(\lambda - \langle A_j,\, w\rangle)^2}\right) A^T + A \,\mathrm{diag}\left(\frac{1}{(\lambda + \langle A_j,\, w\rangle)^2}\right) A^T \,,$$



(a) (b)

(c) (d)

(e) (f)

Fig. 4. Comparison of Newton sketch with various other algorithms, including Newton's Method, Truncated Newton, Stochastic Average Gradient and BFGS in the logistic regression problem with data generated from a Student's t-distribution (see text for details). Plots on the left show log optimality gap versus iteration number, and plots on the right show the log optimality gap versus wall-clock time (bottom). (a), (b): No correlation $\rho = 0$. (c), (d) Correlation $\rho = 0.5$. (e), (f) Correlation $\rho = 0.9$. Newton sketch performs well even with high correlations and non-Gaussian data while first order algorithms perform poorly.

Consequently, we can keep the first term in the Hessian, τI exact and apply partial sketching to the Hessians of the last two terms via

$$S \,\mathrm{diag}\left(\frac{1}{|\lambda - \langle A_j,\, w\rangle|} + \frac{1}{|\lambda + \langle A_j,\, w\rangle|}\right) A^T\,.$$

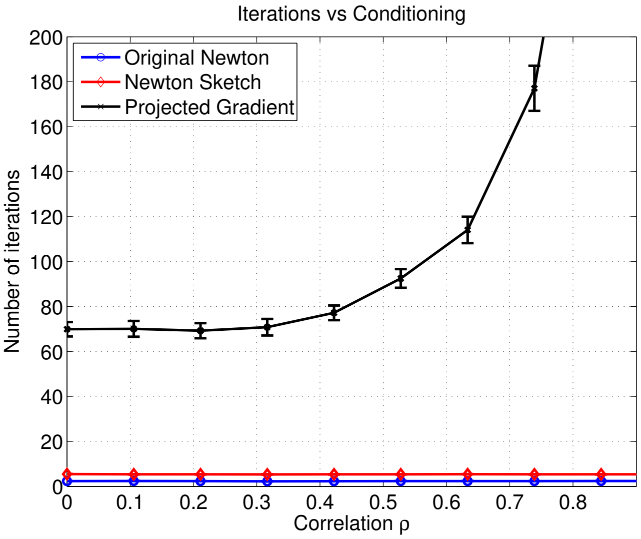


Fig. 5. The performance of Newton Sketch is independent of condition numbers and problem related quantities. Plots of the number of iterations required to reach 10^{-6} accuracy in ℓ_1-constrained logistic regression using Newton's Method and Projected Gradient Descent using line search.

Since the partially sketched Hessian is of the form $tI_n + VV^T$, where V is rank at most m, we can use matrix inversion lemma for efficiently calculating Newton Sketch updates. The complexity of the above strategy for $d > n$ is $\mathcal{O}(nm^2)$, where m is at most n, whereas traditional interior point solvers are typically $\mathcal{O}(dn^2)$ per iteration.

In order to test this algorithm, we generated a feature matrix $A \in \mathbb{R}^{n\times d}$ with $d = 4096$ features and $n = 50$ observations. Each row $a_i \in \mathbb{R}^d$ was generated from the multivariate Gaussian distribution $N(0, \Sigma)$ with $\Sigma_{ij} = 2 * |0.5|^{i-j}$. For a given problem instance, we ran 10 independent trials of the sketched barrier method with $m = 4d$ and ROS sketch, and compared the results to the original barrier method. Figure 6 shows the the duality gap versus iteration number (top panel) and versus the wall-clock time (bottom panel) for the original barrier method (blue) and sketched barrier method (red): although the sketched algorithm requires more iterations, these iterations are cheaper, leading to a smaller wall-clock time. This point is reinforced by Figure 7, where we plot the wall-clock time required to reach a duality gap of 10^{-6} versus the number of features n in problem families of increasing size. Note that the sketched barrier method outperforms the original barrier method, with significantly less computation time for obtaining similar accuracy.

6. Proofs.

We now turn to the proofs of our theorems, with more technical details deferred to the appendices.

6.1. Proof of Theorem 3.1.

For any vector $x \in \mathrm{dom}\,(f)$, and vector $r \in \mathbb{R}^d\backslash\{0\}$, we define the following pair of random variables

$$\begin{aligned} Z_u(S;\, x, r) &:= \sup_{w \in \{\nabla^2 f(x)^{1/2}\mathcal{K}\}\cap\mathcal{S}^{n-1}} \langle w,\, \big(S^TS - I\big)\frac{r}{\|r\|_2}\rangle, \\ Z_\ell(S;\, x) &:= \inf_{w \in \{\nabla^2 f(x)^{1/2}\mathcal{K}\}\cap\mathcal{S}^{n-1}} \|Sw\|_2^2. \end{aligned}$$

Of particular interest to us in analyzing the sketched Newton updates are the sequence of random variables

$$Z_u^t := Z_u(S^t;\, x^t, \nabla^2 f(x^t)^{1/2}\Delta^t), \quad \text{and} \quad Z_\ell^t := Z_\ell(S^t;\, x^t),$$

as defined by the iterates $\{x^t\}_{t=0}^\infty$ and sketching matrices $\{S^t\}_{t=0}^\infty$ of the algorithm.

For a given iteration t and tolerance parameter $\epsilon \in (0, \frac{2\gamma}{9\beta}]$, we define the "good event"

$$\mathcal{E}^t := \Big\{ Z_\ell^t \leq \frac{\epsilon}{2}, \text{ and } Z_u^t \geq 1 - \epsilon \Big\}. \tag{6.1}$$

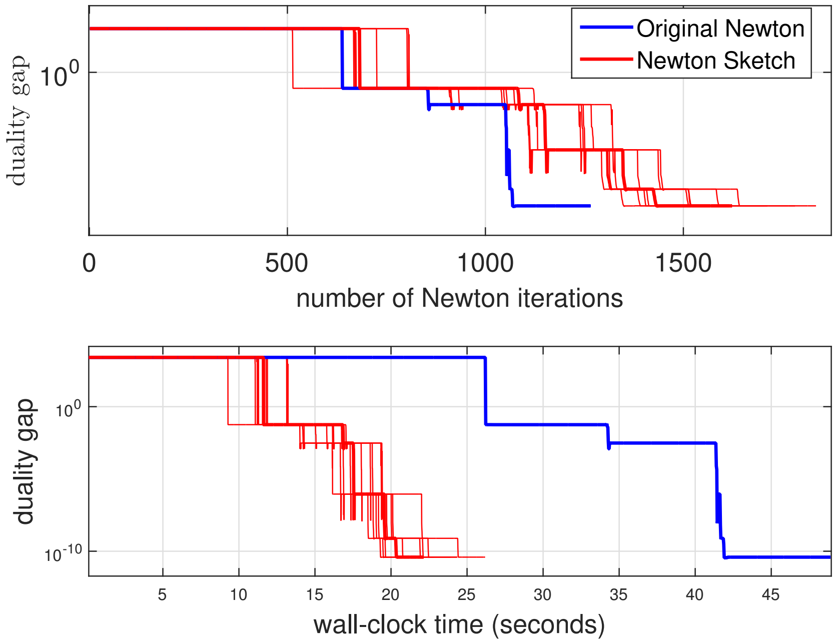


Fig. 6. Plots of the duality gap versus iteration number (top panel) and duality gap versus wall-clock time (bottom panel) for the original barrier method (blue) and sketched barrier method (red). The sketched interior point method is run 10 times independently yielding slightly different curves in red. While the sketched method requires more iterations, its overall wall-clock time is much smaller.

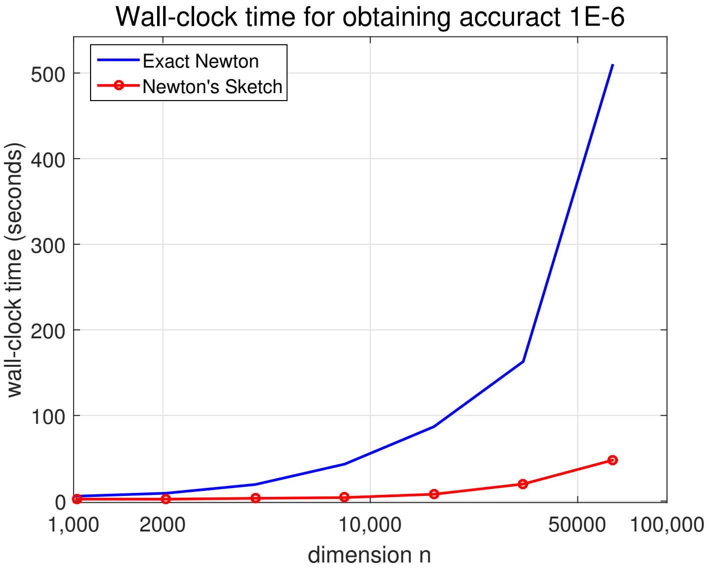


Fig. 7. Plot of the wall-clock time in seconds for reaching a duality gap of 10^{-6} for the standard and sketched interior point methods as n increases (in log-scale). The sketched interior point method has significantly lower computation time compared to the original method.

Given these two definitions, the proof of the theorem is based on two auxiliary lemmas, the first of which establishes a key recursion on the error in the algorithm:

Lemma 6.1 (Key recursion). *Suppose that the event $\cap_{t=1}^{N}\mathcal{E}^t$ holds. Then given any initialization x^0 such that $\|x^0 - x^*\|_2 \leq \frac{\gamma}{8L}$, the error vectors $\Delta^t = x^t - x^*$ satisfy the recursion*

$$\|\Delta^{t+1}\|_2 \leq \frac{Z_u^t}{Z_\ell^t}\,\frac{\beta}{7\gamma}\|\Delta^t\|_2 + \frac{1}{Z_\ell^t}\,\frac{8L}{7\gamma}\|\Delta^t\|_2^2 \qquad \textit{for all } t = 0, 1, \ldots, N-1. \tag{6.2}$$

Note that since we have $\frac{Z_u^t}{Z_\ell^t} \leq \epsilon$ and $\frac{1}{Z_\ell^t} \leq 2$ whenever the event $\cap_{t=1}^N \mathcal{E}^t$ holds, the bound (3.11) stated in the theorem then follows.

In order to complete the proof, we need to establish that the event $\cap_{t=1}^N \mathcal{E}^t$ holds with the stated probability. The following result gives sufficient conditions on the sketch dimension for such a guarantee to hold:

Lemma 6.2 (Sufficient conditions on sketch dimension [31]).

(a) For sub-Gaussian sketch matrices, given a sketch size $m > \frac{c_0}{\epsilon^2} \max_{x \in \mathcal{C}} \mathcal{W}^2(\nabla^2 f(x)^{1/2}\mathcal{K})$, *we have*

$$\mathbb{P}\big[\mathcal{E}^t\big] \geq 1 - c_1 e^{-c_2 m \epsilon^2}. \tag{6.3}$$

(b) For randomized orthogonal system (ROS) sketches and JL embeddings, over the class of self-bounding cones, given a sketch size $m > \frac{c_0 \, \log^4 n}{\epsilon^2} \max_{x \in \mathcal{C}} \mathcal{W}^2(\nabla^2 f(x)^{1/2}\mathcal{K})$, *we have*

$$\mathbb{P}\big[\mathcal{E}^t\big] \geq 1 - c_1 e^{-c_2 \frac{m\epsilon^2}{\log^4 n}}. \tag{6.4}$$

Together with Lemma 6.1, the claim of the theorem follows.

It remains to prove Lemma 6.1, and the bulk of our effort is devoted to this task.

Proof of Lemma 6.1: We prove the recursion (6.2) by exploiting the convex optimality conditions that define the iterates x^{t+1} and the optimum x^*. Recall the function $x \mapsto \Phi(x; S^t)$ that underlies the sketch Newton update (3.2) in moving from iterate x^t to iterate x^{t+1}. Since the vectors x^{t+1} and x^* are optimal and feasible, respectively, for the constrained optimization problem, the error vector $\Delta^{t+1} := x^{t+1} - x^*$ satisfies the inequality $\langle \nabla\Phi(x^{t+1}; S^t), \, -\Delta^{t+1} \rangle \geq 0$, or equivalently

$$\langle (S^t \nabla^2 f(x^t)^{1/2})^T S^t \nabla^2 f(x^t)^{1/2} (\Delta^{t+1} - \Delta^t) + \nabla f(x^t), \, -\Delta^{t+1} \rangle \geq 0.$$

Similarly, since x^* and x^{t+1} are optimal and feasible, respectively, for the minimization of f, we have

$$\langle f(x^*), \, \Delta^{t+1} \rangle \geq 0.$$

Adding these two inequalities and re-arranging leads to the *basic inequality*

$$\underbrace{\|S^t \nabla^2 f(x^t)^{1/2} \Delta^{t+1}\|_2^2}_{\text{LHS}} \leq \underbrace{\langle S^t \nabla^2 f(x^t)^{1/2} \Delta^{t+1}, \, S^t \nabla^2 f(x^t)^{1/2} \Delta^t \rangle - \langle \nabla f(x^t) - \nabla f(x^*), \, \Delta^{t+1} \rangle}_{\text{RHS}} \tag{6.5}$$

This inequality forms the core of our argument: in particular, the next steps in our proof are devoted to establishing the following bounds on the left-hand and right-hand sides:

$$\text{LHS} \geq Z_\ell^t \Big\{ \gamma - L\|\Delta^t\|_2 \Big\} \|\Delta^{t+1}\|_2^2, \quad \text{and} \tag{6.6a}$$

$$\text{RHS} \leq Z_u^t \Big\{ \beta + L\|\Delta^t\|_2 \Big\} \|\Delta^t\|_2 \|\Delta^{t+1}\|_2 + L\|\Delta^t\|_2^2 \|\Delta^{t+1}\|_2. \tag{6.6b}$$

Taking these bounds as given for the moment, let us complete the proof of the recursion (6.2). Our proof consists of two steps:

- we first show that bound (6.2) holds for Δ^{t+1} whenever $\|\Delta^t\|_2 \leq \frac{\gamma}{8L}$.
- we then show by induction that, conditioned on the event $\cap_{t=1}^N \mathcal{E}^t$, the bound $\|\Delta^t\|_2 \leq \frac{\gamma}{8L}$ holds for all iterations $t = 0, 1, \ldots, N$.

Assuming that $\|\Delta^t\|_2 \leq \frac{\gamma}{8L}$, then our basic inequality (6.5) combined with the bounds (6.6) implies that

$$\|\Delta^{t+1}\|_2 \leq \frac{Z_u^t \{\beta + L\|\Delta^t\|_2\}}{Z_\ell^t \{\gamma - L\|\Delta^t\|_2\}} \|\Delta^t\|_2 + \frac{L}{Z_\ell^t \{\gamma - L\|\Delta^t\|_2\}} \|\Delta^t\|_2^2.$$

We have $L\|\Delta^t\|_2 \leq \gamma/8 \leq \beta/8$, and $(\gamma - L\|\Delta^t\|_2)^{-1} \leq \frac{8}{7\gamma}$ hence

$$\|\Delta^{t+1}\|_2 \leq \frac{Z_u^t}{Z_\ell^t} \, \frac{9}{7} \frac{\beta}{\gamma} \|\Delta^t\|_2 + \frac{1}{Z_\ell^t} \frac{8L}{7\gamma} \|\Delta^t\|_2^2, \tag{6.7}$$

thereby verifying the claim (6.2).

Now we need to check for any iteration t, the bound $\|\Delta^t\|_2 \le \frac{\gamma}{8L}$ holds. We do so by induction. The base case is trivial since $\|\Delta^0\|_2 \le \frac{\gamma}{8L}$ by assumption. Supposing that the bound holds at time t, by our argument above, inequality (6.7) holds, and hence

$$\|\Delta^{t+1}\|_2 \le \frac{9}{56}\frac{\beta Z_u^t}{L\, Z_\ell^t} + \frac{16L}{7\gamma Z_\ell^t}\frac{\gamma^2}{64L^2} \;=\; \frac{Z_u^t}{Z_\ell^t}\frac{9}{28}\frac{\beta}{L} + \frac{1}{Z_\ell^t}\frac{1}{28}\frac{\gamma}{L}.$$

Whenever $\mathcal{E}^t$ holds, we have $\frac{Z_u^t}{Z_\ell^t} \le \frac{2\gamma}{9\beta}$ and $\frac{1}{Z_\ell^t} \le \frac{1}{2}$, whence $\|\Delta^{t+1}\|_2 \le \left(\frac{1}{28} + \frac{1}{14}\right)\frac{\gamma}{L} \le \frac{\gamma}{8L}$, as claimed.

The final remaining detail is to prove the bounds (6.6).

Proof of the lower bound (6.6a). We first prove the lower bound (6.6a) on the LHS. Since $\nabla^2 f(x^t)^{1/2}\Delta^{t+1} \in \nabla^2 f(x^t)^{1/2}\mathcal{K}$, the definition of Z_ℓ^t ensures that

$$\begin{aligned}
\text{LHS} \;=\; \|S^t\nabla^2 f(x^t)^{1/2}\Delta^{t+1}\|_2^2 &\ge Z_\ell^t\|\nabla^2 f(x^t)^{1/2}\Delta^{t+1}\|_2^2 \\
&\overset{(i)}{=} Z_\ell^t(\Delta^{t+1})^T\nabla^2 f(x^t)\Delta^{t+1} \\
&= Z_\ell^t\{(\Delta^{t+1})^T\nabla^2 f(x^*)\Delta^{t+1} + (\Delta^{t+1})^T(\nabla^2 f(x^t) - \nabla^2 f(x^*))\Delta^{t+1} \\
&\overset{(ii)}{\ge} Z_\ell^t\left\{\gamma\|\Delta^{t+1}\|_2^2 - L\|\Delta^{t+1}\|_2^2\|\Delta^t\|_2\right\}
\end{aligned}$$

where step (i) follows since $(\nabla^2 f(x)^{1/2})^T\nabla^2 f(x)^{1/2} = \nabla^2 f(x)$, and step (ii) follows from the definitions of γ and L.

Proof of the upper bound (6.6b). Next we prove the upper bound (6.6b) on the RHS. Throughout this proof, we write S instead of S^t so as to simplify notation. By the integral form of Taylor series, we have

$$\begin{aligned}
\text{RHS} &= \int_0^1 (\Delta^t)^T\left[(S\nabla^2 f(x^t)^{1/2})^T S\nabla^2 f(x^t)^{1/2} - \nabla^2 f(x^t + u(x^* - x^t))\right]\Delta^{t+1}du \\
&= T_1 + T_2
\end{aligned}$$

where

$$T_1 := (\Delta^t)^T\left[(S\nabla^2 f(x^t)^{1/2})^T S\nabla^2 f(x^t)^{1/2} - \nabla^2 f(x^t)\right]\Delta^{t+1}, \quad \text{and} \tag{6.8a}$$

$$T_2 := \int_0^1 (\Delta^t)^T\left[-\nabla^2 f(x^t + u(x^* - x^t)) + \nabla^2 f(x^t)\right]\Delta^{t+1}du. \tag{6.8b}$$

Here the decomposition into T_1 and T_2 follows by adding and subtracting the term $(\Delta^t)^T\nabla^2 f(x^t)\Delta^{t+1}$.

We begin by upper bounding the term T_1. By the definition of Z_u^t, we have

$$T_1 \le \left|(\Delta^t)^T Q^T(x^t)\left[\frac{S^TS}{m} - I\right]\nabla^2 f(x^t)^{1/2}\Delta^{t+1}\right| \le Z_2\|\nabla^2 f(x^t)^{1/2}\Delta^t\|_2\|\nabla^2 f(x^t)^{1/2}\Delta^{t+1}\|_2.$$

By adding and subtracting terms, we have

$$\begin{aligned}
\|\nabla^2 f(x^t)^{1/2}\Delta^t\|_2^2 = (\Delta^t)^T\nabla^2 f(x^t)\Delta^t &= (\Delta^t)^T\nabla^2 f(x^*)\Delta^t + (\Delta^t)^T\left[\nabla^2 f(x^t) - \nabla^2 f(x^*)\right]\Delta^t \\
&\le \beta\|\Delta^t\|_2^2 + L\|\Delta^t\|^3 = \|\Delta^t\|_2^2(\beta + L\|\Delta^t\|),
\end{aligned}$$

where the final step follows from the definitions of β and L, as bounds on the Hessian, and its Lipschitz constant, respectively. A similar argument yields

$$\|\nabla^2 f(x^t)^{1/2}\Delta^{t+1}\|_2^2 \le \|\Delta^{t+1}\|_2^2(\beta + L\|\Delta^t\|).$$

Overall, we have shown that

$$T_1 \le Z_u^t(\beta + L\|\Delta^t\|)\|\Delta^t\|_2\|\Delta^{t+1}\|_2. \tag{6.9}$$

Turning to the quantity T_2, we have

$$
\begin{aligned}
T_2 &\leq \left\{ \int_0^1 \sup_{v, \tilde{v} \in \mathcal{K} \cap \mathcal{S}^{d-1}} \left| v^T \left[\nabla^2 f(x^t + u(x^* - x^t)) - \nabla^2 f(x^t) \right] \tilde{v} \right| du \right\} \|\Delta^t\|_2 \|\Delta^{t+1}\|_2 \\
&\leq L \|\Delta^t\|_2^2 \|\Delta^{t+1}\|_2, \qquad (6.10)
\end{aligned}
$$

where the final step uses the local Lipschitz property again. Combining the bound (6.9) with the bound (6.10) yields the bound (6.6b) on the RHS.

6.2. Proof of Theorem 4.1. Recall that in this case, we assume that f is a self-concordant and strictly convex function. We adopt the following notation and conventions from Nesterov and Nemirovski [28]. For a given vector $x \in \mathbb{R}^d$, we define the pair of dual norms

$$
\|u\|_x := \langle \nabla^2 f(x) u, \, u \rangle^{1/2}, \quad \text{and} \quad \|v\|_x^* := \langle \nabla^2 f(x)^{-1} v, \, v \rangle^{1/2},
$$

as well as the Newton decrement

$$
\lambda_f(x) = \langle \nabla^2 f(x)^{-1} \nabla f(x), \, \nabla f(x) \rangle^{1/2} = \|\nabla^2 f(x)^{-1} \nabla f(x)\|_x \; = \; \|\nabla^2 f(x)^{-1/2} \nabla f(x)\|_2 \, .
$$

Note that $\nabla^2 f(x)^{-1}$ is well-defined for strictly convex self-concordant functions. In terms of this notation, the exact Newton update is given by $x \mapsto x_{\text{NE}} := x + v_{\text{NE}}$, where

$$
v_{\text{NE}} := \arg \min_{z \in \mathcal{C} - x} \Big\{ \underbrace{\frac{1}{2} \|\nabla^2 f(x)^{1/2} z\|_2^2 + \langle z, \, \nabla f(x) \rangle}_{\Phi(z)} \Big\}. \tag{6.11a}
$$

On the other hand, the Newton sketch update is given by $x \mapsto x_{\text{NSK}} := x + v_{\text{NSK}}$, where

$$
v_{\text{NSK}} := \arg \min_{z \in \mathcal{C} - x} \left\{ \frac{1}{2} \|S \nabla^2 f(x)^{1/2} z\|_2^2 + \langle z, \, \nabla f(x) \rangle \right\}. \tag{6.11b}
$$

The proof of Theorem 4.1 given in this section involves the unconstrained case ($\mathcal{C} = \mathbb{R}^d$), whereas the proofs of later theorems involve the more general constrained case. In the unconstrained case, the two updates take the simpler forms

$$
x_{\text{NE}} = x - (\nabla^2 f(x))^{-1} \nabla f(x), \qquad \text{and} \qquad x_{\text{NSK}} = x - (\nabla^2 f(x)^{1/2} S^T S \nabla^2 f(x)^{1/2})^{-1} \nabla f(x).
$$

Let us make a few high-level remarks about the structure of the proof. For a self-concordant function, the sub-optimality of the Newton iterate x_{NE} in function value satisfies the bound

$$
f(x_{\text{NE}}) - \underbrace{\min_{x \in \mathbb{R}^d} f(x)}_{f(x^*)} \leq \left[\lambda_f(x_{\text{NE}}) \right]^2 \, .
$$

This classical bound is not directly applicable to the Newton sketch update, since it involves the *approximate* Newton decrement $\widetilde{\lambda}_f(x)^2 = -\langle \nabla f(x), \, v_{\text{NSK}} \rangle$, as opposed to the *exact one* $\lambda_f(x)^2 = -\langle \nabla f(x), \, v_{\text{NE}} \rangle$. Thus, our strategy is to prove that with high probability over the randomness in the sketch matrix, the approximate Newton decrement can be used as an exit condition.

Our argument for doing so consists of two main parts. First, we show that under appropriate conditions on the random sketch, the approximate and exact Newton updates are guaranteed to be relatively close. Second, in analogy to the analysis of classical Newton's method, we analyze the sketched Newton algorithm in two phases, depending on whether the sketched Newton decrement $\widetilde{\lambda}_f(x)$ is strictly greater than or less than or equal to η.

6.2.1. Relating the sketched and exact Newton directions. Recalling the definitions (6.11a) and (6.11b) of the exact v_{NE} and sketched Newton v_{NSK} update directions, our first step is to show that under suitable conditions on the random sketch, these two directions are close with high probability. Recall the definition (3.8) of the tangent cone $\mathcal{K}$ of the set $\mathcal{C}$ at a given vector $x \in \mathcal{C}$. With this notation, the following lemma provides a high probability bound on these two directions:

LEMMA 6.3. *Let $S \in \mathbb{R}^{m \times n}$ be a sub-Gaussian, ROS or JL sketching matrix and consider any fixed vector $x \in \mathcal{C}$ independent of the sketch matrix. If $m \geq c_0 \frac{\mathcal{W}(\nabla^2 f(x)^{1/2}\mathcal{K})^2}{\epsilon^2}$, then*

$$\left\| \nabla^2 f(x)^{1/2}(v_{\scriptscriptstyle NSK} - v_{\scriptscriptstyle NE}) \right\|_2 \leq \epsilon \left\| \nabla^2 f(x)^{1/2} v_{\scriptscriptstyle NE} \right\|_2 \tag{6.12}$$

with probability at least $1 - c_1 e^{-c_2 m \epsilon^2}$.

Proof. The proof follows similar lines to the proof of Theorem 3.1, in particular exploiting the convex optimality conditions that define the exact and sketched Newton updates. Let u be a unit-norm vector independent of S, and consider the random quantities

$$Z_\ell(S, x) := \inf_{v \in \nabla^2 f(x)^{1/2} \mathcal{K}^t \cap \mathcal{S}^{n-1}} \|Sv\|_2^2 \quad \text{and} \tag{6.13a}$$

$$Z_u(S, x) := \sup_{v \in \nabla^2 f(x)^{1/2} \mathcal{K}^t \cap \mathcal{S}^{n-1}} \left| \langle u, \, (S^T S - I_n) \, v \rangle \right|. \tag{6.13b}$$

By the optimality and feasibility of v_{NSK} and v_{NE} (respectively) for the sketched Newton update (6.11b), we have

$$\frac{1}{2} \|S \nabla^2 f(x)^{1/2} v_{\text{NSK}}\|_2^2 - \langle v_{\text{NSK}}, \, \nabla f(x) \rangle \leq \frac{1}{2} \|\nabla^2 f(x)^{1/2} v_{\text{NE}}\|_2^2 - \langle v_{\text{NE}}, \, \nabla f(x) \rangle.$$

Defining the difference vector $\widehat{e} := v_{\text{NSK}} - v_{\text{NE}}$, some algebra leads to the basic inequality

$$\frac{1}{2} \|S \nabla^2 f(x)^{1/2} \widehat{e}\|_2^2 \leq -\langle \nabla^2 f(x)^{1/2} v_{\text{NE}}, \, S^T S \nabla^2 f(x)^{1/2} \widehat{e} \rangle + \langle \widehat{e}, \, \nabla f(x) \rangle. \tag{6.14}$$

Moreover, by the optimality and feasibility of v_{NE} and v_{NSK} for the exact Newton update (6.11a), we have

$$\langle \nabla^2 f(x) v_{\text{NE}} - \nabla f(x), \, \widehat{e} \rangle \; = \; \langle \nabla^2 f(x) v_{\text{NE}} - \nabla f(x), \, v_{\text{NSK}} - v_{\text{NE}} \rangle \geq 0. \tag{6.15}$$

Consequently, by adding and subtracting $\langle \nabla^2 f(x) v_{\text{NE}}, \, \widehat{e} \rangle$, we find that

$$\frac{1}{2} \|S \nabla^2 f(x)^{1/2} \widehat{e}\|_2^2 \leq \left| \langle \nabla^2 f(x)^{1/2} v_{\text{NE}}, \, \big(I_n - S^T S\big) \nabla^2 f(x)^{1/2} \widehat{e} \rangle \right|. \tag{6.16}$$

By definition, the error vector $\widehat{e}$ belongs to the cone $\mathcal{K}^t$ and the vector $\nabla^2 f(x)^{1/2} v_{\text{NE}}$ is fixed and independent of the sketch. Consequently, invoking definitions (6.13a) and (6.13b) of the random variables Z_ℓ and Z_u yields

$$\frac{1}{2} \|S \nabla^2 f(x)^{1/2} \widehat{e}\|_2^2 \geq \frac{Z_\ell}{2} \|\nabla^2 f(x)^{1/2} \widehat{e}\|_2^2,$$
$$\left| \langle \nabla^2 f(x)^{1/2} v_{\text{NE}}, \, \big(I_n - S^T S\big) \nabla^2 f(x)^{1/2} \widehat{e} \rangle \right| \leq Z_u \|\nabla^2 f(x)^{1/2} v_{\text{NE}}\|_2 \, \|\nabla^2 f(x)^{1/2} \widehat{e}\|_2,$$

Putting together the pieces, we find that

$$\left\| \nabla^2 f(x)^{1/2}(v_{\text{NSK}} - v_{\text{NE}}) \right\|_2 \leq \frac{2 Z_u(S, x)}{Z_\ell(S, x)} \left\| \nabla^2 f(x)^{1/2}(v_{\text{NE}}) \right\|_2 \; . \tag{6.17}$$

Finally, for any $\delta \in (0, 1)$, let us define the event $\mathcal{E}(\delta) = \{Z_\ell \geq 1 - \delta, \quad \text{and} \quad Z_u \leq \delta\}$. By Lemma 4 and Lemma 5 from our previous paper [31], we are guaranteed that $\mathbb{P}[\mathcal{E}(\delta)] \geq 1 - c_1 e^{-c_2 m \delta^2}$. Conditioned on the event $\mathcal{E}(\delta)$, the bound (6.17) implies that

$$\left\| \nabla^2 f(x)^{1/2}(v_{\text{NSK}} - v_{\text{NE}}) \right\|_2 \leq \frac{2\delta}{1 - \delta} \left\| \nabla^2 f(x)^{1/2}(v_{\text{NE}}) \right\|_2 \; .$$

By setting $\delta = \frac{\epsilon}{4}$, the claim follows.
□

6.2.2. Two phase analysis. Similar to the standard analysis of Newton's method, our next step in the analysis of the Newton sketch algorithm is to consider two different phases, as defined by the magnitude of the decrement $\widetilde{\lambda}_f(x)$:

- **First phase:** Decrement $\widetilde{\lambda}_f(x) > \eta$.
- **Second phase:** Decrement $\widetilde{\lambda}_f(x) \leq \eta$.

In each phase, we establish different guarantees on the behavior of the sketched update x_{NSK} based on a starting vector x.

Lemma 6.4. *For $\epsilon \in (0, 1/2)$, there exist constants $\nu > 0$ and $\eta \in (0, 1/16)$ such that:*

(a) If $\widetilde{\lambda}_f(x) > \eta$, then $f(x_{NSK}) - f(x) \leq -\nu$ with probability at least $1 - c_1 e^{-c_2 m\epsilon^2}$.

(b) Conversely, if $\widetilde{\lambda}_f(x) \leq \eta$, then

$$\widetilde{\lambda}_f(x_{NSK}) \leq \widetilde{\lambda}_f(x), \quad \textit{and} \tag{6.18a}$$

$$\lambda_f(x_{NSK}) \leq (\frac{16}{25})\lambda_f(x), \tag{6.18b}$$

where both bounds hold with probability $1 - c_1 e^{c_2 m\epsilon^2}$.

Using this lemma, let us now complete the proof of the theorem, dividing our analysis into the two phases of the algorithm.

First phase analysis: Lemma 6.4(a) ensures that each iteration in the first phase decreases the function value by at least $\nu > 0$; consequently, the number of first phase iterations N_1 is at most

$$N_1 := \frac{f(x^0) - f(x^*)}{\nu},$$

with probability at least $1 - N_1 c_1 e^{-c_2 m}$.

Second phase analysis: Next, let us suppose that at some iteration t, the condition $\widetilde{\lambda}_f(x^t) \leq \eta$ holds, so that part (b) of Lemma 6.4 can be applied. In fact, the bound (6.18a) then guarantees that $\widetilde{\lambda}_f(x^{t+1}) \leq \eta$, so that we may apply the contraction bound (6.18b) repeatedly for N_2 rounds so as to obtain that

$$\lambda_f(x^{t+N_2}) \leq (\frac{16}{25})^{N_2} \lambda_f(x^t)$$

with probability $1 - N_2 c_1 e^{c_2 m}$.

Since $\lambda_f(x^t) \leq \eta \leq 1/16$ by assumption, the self-concordance of f then implies that

$$f(x^{t+k}) - f(x^*) \leq \left(\frac{16}{25}\right)^k \frac{1}{16}.$$

Therefore, in order to ensure that and consequently for achieving $f(x^{t+k}) - f(x^*) \leq \epsilon$, it suffices for the number of second phase iterations to be lower bounded as $N_2 \geq 0.65 \log_2(\frac{1}{16\epsilon})$.

Putting together the two phases, we conclude that the total number of iterations N required to achieve ϵ- accuracy is at most

$$N = N_1 + N_2 \; \leq \; \frac{f(x^0) - f(x^*)}{\gamma} + 0.65 \log_2(\frac{1}{16\epsilon}),$$

and moreover, this guarantee holds with probability at least $1 - N c_1 e^{-c_2 m\epsilon^2}$.

Thus, in order to complete our proof of the theorem, theorem, it remains to prove Lemma 6.4, and we do in the next two subsections.

6.2.3. Proof of Lemma 6.4(a). Our proof of this part is performed conditionally on the event $\mathcal{D} := \{\widetilde{\lambda}_f(x) > \eta\}$. Our strategy is to show that the backtracking line search leads to a stepsize $s > 0$ such that function decrement in moving from the current iterate x to the new sketched iterate $x_{\text{NSK}} = x + sv_{\text{NSK}}$ is at least

$$f(x_{\text{NSK}}) - f(x) \leq -\nu \quad \text{with probability at least } 1 - c_1 e^{-c_2 m}. \tag{6.19}$$

The outline of our proof is as follows. Defining the univariate function $g(u) := f(x + uv_{\text{NSK}})$ and $\epsilon' = \frac{2\epsilon}{1-\epsilon}$, we first show that the step size choice $\widehat{u} := \frac{1}{1+(1+\epsilon')\widetilde{\lambda}_f(x)}$ satisfies the bound

$$g(\widehat{u}) \leq g(0) - a\widehat{u}\widetilde{\lambda}_f(x)^2, \tag{6.20a}$$

which implies that $\widehat{u}$ satisfies the exit condition of backtracking line search. Therefore, the stepsize s must be lower bounded as $s \geq b\widehat{u}$, which then implies that the updated solution $x_{\text{NSK}} = x + sv_{\text{NSK}}$ satisfies the decrement bound

$$f(x_{\text{NSK}}) - f(x) \leq -ab\frac{\widetilde{\lambda}_f(x)^2}{1 + (1 + \frac{2\epsilon}{1-\epsilon})\widetilde{\lambda}_f(x)}. \tag{6.20b}$$

Since $\widetilde{\lambda}_f(x) > \eta$ by assumption and the function $u \rightarrow \frac{u^2}{1+(1+\frac{2\epsilon}{1-\epsilon})u}$ is monotone increasing, this bound implies that inequality (6.19) holds with $\nu = ab\frac{\eta^2}{1+(1+\frac{2\epsilon}{1-\epsilon})\eta}$.

In order to prove the claims (6.20a) and (6.20b), we first state and prove an auxiliary lemma:

Lemma 6.5.

(a) For $u \in \operatorname{dom} g \cap \mathbb{R}^+$, we have the decrement bound

$$g(u) \leq g(0) + u\langle \nabla f(x),\, v_{NSK}\rangle - u\|[\nabla^2 f(x)]^{1/2} v_{NSK}\|_2 - \log\big(1 - u\|[\nabla^2 f(x)]^{1/2} v_{NSK}\|_2\big). \tag{6.21a}$$

provided that $u\|[\nabla^2 f(x)]^{1/2} v_{NSK}\|_2 < 1$.

(b) With probability at least $1 - c_1 e^{-c_2 m}$, we have

$$\|[\nabla^2 f(x)]^{1/2} v_{NSK}\|_2^2 \leq \left(\frac{1+\epsilon}{1-\epsilon}\right)^2 \big[\widetilde{\lambda}_f(x)\big]^2. \tag{6.21b}$$

Proof. Let us first prove the bound (6.21a). By construction, the function $g(u) = f(x + uv_{\text{NSK}})$ is strictly convex and self-concordant. Consequently, it satisfies the bound $\frac{d}{du}\left(g''(u)^{-1/2}\right) \leq 1$, whence

$$g''(s)^{-1/2} - g''(0)^{-1/2} \; = \; \int_0^s \frac{d}{du}\left(g''(u)^{-1/2}\right) du \; \leq \; s.$$

or equivalently $g''(s) \leq \frac{g''(0)}{(1-sg''(0)^{1/2})^2}$ for $s \in \operatorname{dom} g \cap [0, g''(0)^{-1/2})$. Integrating this inequality twice yields the bound

$$g(u) \leq g(0) + ug'(0) - ug''(0)^{1/2} - \log(1 - ug''(0)^{1/2})\,.$$

Since $g'(u) = \langle \nabla f(x{+}uv_{\text{NSK}}),\, v_{\text{NSK}}\rangle$ and $g''(u) = \langle v_{\text{NSK}},\, \nabla^2 f(x{+}uv_{\text{NSK}})v_{\text{NSK}}\rangle$, the decrement bound (6.21a) follows.

Turning to the proof of the bound (6.21b), we perform this analysis conditional on the bound (6.12) from Lemma 6.3. We begin by observing that

$$\begin{aligned}\|[\nabla^2 f(x)]^{1/2} v_{\text{NSK}}\|_2 &\leq \|[\nabla^2 f(x)]^{1/2} v_{\text{NE}}\|_2 + \|[\nabla^2 f(x)]^{1/2}(v_{\text{NSK}} - v_{\text{NE}})\|_2 \\ &= \lambda_f(x) + \|[\nabla^2 f(x)]^{1/2}(v_{\text{NSK}} - v_{\text{NE}})\|_2\,.\end{aligned} \tag{6.22}$$

Lemma 6.3 implies that $\|\nabla^2[f(x)]^{1/2}(v_{\text{NSK}} - v_{\text{NE}})\|_2 \leq \epsilon \|\nabla^2[f(x)]^{1/2} v_{\text{NE}}\|_2 = \epsilon \lambda_f(x)$. In conjunction with the bound (6.22), we see that

$$\|[\nabla^2 f(x)]^{1/2} v_{\text{NSK}}\|_2 \leq (1+\epsilon)\lambda_f(x)\,. \tag{6.23}$$

Our next step is to lower bound the term $\langle \nabla f(x),\, v_{\text{NSK}}\rangle$: in particular, by adding and subtracting a factor of the original Newton step v_{NE}, we find that

$$\begin{aligned}
\langle \nabla f(x),\, v_{\text{NSK}}\rangle &= \langle [\nabla^2 f(x)]^{-1/2}\nabla f(x),\, \nabla^2[f(x)]^{1/2} v_{\text{NSK}}\rangle \\
&= \langle [\nabla^2 f(x)]^{-1/2}\nabla f(x),\, \nabla^2[f(x)]^{1/2} v_{\text{NE}}\rangle + \langle [\nabla^2 f(x)]^{-1/2}\nabla f(x),\, \nabla^2[f(x)]^{1/2}(v_{\text{NSK}} - v)\rangle \\
&= -\|\nabla^2[f(x)]^{-1/2}\nabla f(x)\|_2^2 + \langle [\nabla^2 f(x)]^{-1/2}\nabla f(x),\, \nabla^2[f(x)]^{1/2}(v_{\text{NSK}} - v_{\text{NE}})\rangle \\
&\leq -\|\nabla^2[f(x)]^{-1/2}\nabla f(x)\|_2^2 + \|[\nabla^2 f(x)]^{-1/2}\nabla f(x)\|_2 \|\nabla^2[f(x)]^{1/2}(v_{\text{NSK}} - v_{\text{NE}})\|_2 \\
&= -\lambda_f(x)^2 + \lambda_f(x)\|\nabla^2[f(x)]^{1/2}(v_{\text{NSK}} - v_{\text{NE}})\|_2 \\
&\leq -\lambda_f(x)^2(1-\epsilon), \qquad (6.24)
\end{aligned}$$

where the final step again makes use of Lemma 6.3. Repeating the above argument in the reverse direction yields the lower bound $\langle \nabla f(x),\, v_{\text{NSK}}\rangle \geq -\lambda_f(x)^2(1+\epsilon)$, so that we may conclude that

$$|\widetilde{\lambda}_f(x) - \lambda_f(x)| \leq \epsilon \lambda_f(x). \tag{6.25}$$

Finally, by squaring both sides of the inequality (6.22) and combining with the above bounds gives

$$\|[\nabla^2 f(x)]^{1/2} v_{\text{NSK}}\|_2^2 \;\leq\; \frac{-(1+\epsilon)^2}{1-\epsilon}\,\langle \nabla f(x),\, v_{\text{NSK}}\rangle = \frac{(1+\epsilon)^2}{1-\epsilon}\,\widetilde{\lambda}_f^2(x) \;\leq\; \left(\frac{1+\epsilon}{1-\epsilon}\right)^2 \widetilde{\lambda}_f^2(x),$$

as claimed. □

We are now equipped to return to the proofs of our earlier claims (6.20a) and (6.20b). Recalling our shorthand $\epsilon' := \frac{1+\epsilon}{1-\epsilon} - 1 = \frac{2\epsilon}{1-\epsilon}$, substituting inequality (6.21b) into the decrement formula (6.21a) yields

$$\begin{aligned}
g(u) &\leq g(0) - u\widetilde{\lambda}_f(x)^2 - u(1+\epsilon')\,\widetilde{\lambda}_f(x) - \log(1 - u(1+\epsilon')\,\widetilde{\lambda}_f(x)) \qquad (6.26)\\
&= g(0) - \Big\{u(1+\epsilon')^2\widetilde{\lambda}_f(x)^2 + u(1+\epsilon')\,\widetilde{\lambda}_f(x) + \log(1 - u(1+\epsilon')\,\widetilde{\lambda}_f(x))\Big\} \\
&\qquad + u((1+\epsilon')^2 - 1)\widetilde{\lambda}_f(x)^2
\end{aligned}$$

where we added and subtracted $u(1+\epsilon')^2\widetilde{\lambda}_f(x)^2$ so as to obtain the final equality.

We now prove inequality (6.20a). Now since the choice $u = \widehat{u} := \frac{1}{1+(1+\epsilon')\widetilde{\lambda}_f(x)}$ satisfies the conditions of Lemma 6.5, we are guaranteed that

$$g(\widehat{u}) \leq g(0) - (1+\epsilon')\,\widetilde{\lambda}_f(x) + \log(1 + (1+\epsilon')\,\widetilde{\lambda}_f(x)) + \frac{({\epsilon'}^2 + 2\epsilon')\widetilde{\lambda}_f(x)^2}{1 + (1+\epsilon')\widetilde{\lambda}_f(x)}\,.$$

Making use of the standard inequality $-u + \log(1+u) \leq -\frac{\frac{1}{2}u^2}{(1+u)}$ (for instance, see the book [6]), we find that

$$\begin{aligned}
g(\widehat{u}) &\leq g(0) - \frac{\frac{1}{2}(1+\epsilon')^2\widetilde{\lambda}_f(x)^2}{1 + (1+\epsilon')\widetilde{\lambda}_f(x)} + \frac{({\epsilon'}^2 + 2\epsilon')\widetilde{\lambda}_f(x)^2}{1 + (1+\epsilon')\widetilde{\lambda}_f(x)} \\
&= g(0) - (\frac{1}{2} - \frac{1}{2}{\epsilon'}^2 - \epsilon')\widetilde{\lambda}_f(x)^2\widehat{u} \\
&\leq g(0) - \alpha\widetilde{\lambda}_f(x)^2\widehat{u},
\end{aligned}$$

where the final inequality follows from our assumption $\alpha \leq \frac{1}{2} - \frac{1}{2}{\epsilon'}^2 - \epsilon'$. This completes the proof of the bound (6.20a). Finally, the lower bound (6.20b) follows by setting $u = b\widehat{u}$ into the decrement inequality (6.21a). We have thus completed the proof of Lemma 6.4(a).

6.2.4. Proof of Lemma 6.4(b). Our proof of this part hinges on the following auxiliary lemma:

LEMMA 6.6. *For all* $\epsilon \in (0, 1/2)$, *we have*

$$\lambda_f(x_{\scriptscriptstyle NSK}) \leq \frac{(1+\epsilon)\lambda_f^2(x) + \epsilon\lambda_f(x)}{\Big(1 - (1+\epsilon)\lambda_f(x)\Big)^2}, \qquad \textit{and} \tag{6.27a}$$

$$(1-\epsilon)\,\lambda_f(x) \;\leq \widetilde{\lambda}_f(x) \;\leq\; (1+\epsilon)\lambda_f(x)\,, \tag{6.27b}$$

where all bounds hold with probability at least $1 - c_1 e^{-c_2 m\epsilon^2}$.

Proof. We have already proved the bound (6.27b) during our proof of Lemma 6.5—in particular, see equation (6.25). Accordingly, it remains only to prove the inequality (6.27a).

Introducing the shorthand $\widetilde{\lambda} := (1+\epsilon)\lambda_f(x)$, we first claim that the Hessian satisfies the sandwich relation

$$(1 - s\alpha)^2 \nabla^2 f(x) \preceq \nabla^2 f(x + s v_{\text{NSK}}) \preceq \frac{1}{(1-s\alpha)^2} \nabla^2 f(x)\,, \tag{6.28}$$

for $|1 - s\alpha| < 1$ where $\alpha = (1+\epsilon)\lambda_f(x)$, with probability at least $1 - c_1 e^{-c_2 m\epsilon^2}$. Let us recall Theorem 4.1.6 of Nesterov [27]: it guarantees that

$$(1 - s\|v_{\text{NSK}}\|_x)^2 \nabla^2 f(x) \preceq \nabla^2 f(x + s v_{\text{NSK}}) \preceq \frac{1}{(1 - s\|v_{\text{NSK}}\|_x)^2} \nabla^2 f(x)\;. \tag{6.29}$$

Now recall the bound (6.12) from Lemma 6.3: combining it with an application of the triangle inequality (in terms of the semi-norm $\|v\|_x = \|\nabla^2 f(x)^{1/2} v\|_2$) yields

$$\left\|\nabla^2 f(x)^{1/2} v_{\text{NSK}}\right\|_2 \leq (1+\epsilon)\left\|\nabla^2 f(x)^{1/2} v_{\text{NE}}\right\|_2 \;=\; (1+\epsilon)\|v_{\text{NE}}\|_x\,,$$

with probability at least $1 - e^{-c_1 m\epsilon^2}$, and substituting this inequality into the bound (6.29) yields the sandwich relation (6.28) for the Hessian.

Using this sandwich relation (6.28), the Newton decrement can be bounded as

$$\begin{aligned}
\lambda_f(x_{\text{NSK}}) &= \|\nabla^2 f(x_{\text{NSK}})^{-1/2} \nabla f(x_{\text{NSK}})\|_2 \\
&\leq \frac{1}{(1-(1+\epsilon)\lambda_f(x))} \|\nabla^2 f(x)^{-1/2} \nabla f(x_{\text{NSK}})\|_2 \\
&= \frac{1}{(1-(1+\epsilon)\lambda_f(x))} \left\|\nabla^2 f(x)^{-1/2}\left(\nabla f(x) + \int_0^1 \nabla^2 f(x + s v_{\text{NSK}}) v_{\text{NSK}}\, ds\right)\right\|_2 \\
&= \frac{1}{(1-(1+\epsilon)\lambda_f(x))} \left\|\nabla^2 f(x)^{-1/2}\left(\nabla f(x) + \int_0^1 \nabla^2 f(x + s v_{\text{NSK}}) v_{\text{NE}}\, ds + \Delta\right)\right\|_2,
\end{aligned}$$

where we have defined $\Delta = \int_0^1 \nabla^2 f(x + s v_{\text{NSK}})\,(v_{\text{NSK}} - v_{\text{NE}})\, ds$. By the triangle inequality, we can write $\lambda_f(x_{\text{NSK}}) \leq \frac{1}{(1-(1+\epsilon)\lambda_f(x))}\big(M_1 + M_2\big)$, where

$$M_1 := \left\|\nabla^2 f(x)^{-1/2}\left(\nabla f(x) + \int_0^1 \nabla^2 f(x + t v_{\text{NSK}}) v_{\text{NE}} dt\right)\right\|_2, \quad \text{and} \quad M_2 := \left\|\nabla^2 f(x)^{-1/2}\Delta\right\|_2.$$

In order to complete the proof, it suffices to show that

$$M_1 \leq \frac{(1+\epsilon)\lambda_f(x)^2}{1-(1+\epsilon)\lambda_f(x)}, \quad \text{and} \quad M_2 \leq \frac{\epsilon\lambda_f(x)}{1-(1+\epsilon)\lambda_f(x)}.$$

Bound on M_1: Re-arranging and then invoking the Hessian sandwich relation (6.28) yields

$$\begin{aligned}
M_1 &= \left\| \int_0^1 \left(\nabla^2 f(x)^{-1/2} \nabla^2 f(x + s v_{\text{NSK}}) \nabla^2 f(x)^{-1/2} - I \right) ds \, \left(\nabla^2 f(x)^{1/2} v_{\text{NE}} \right) \right\|_2 \\
&\leq \left| \int_0^1 \left(\frac{1}{(1 - s(1+\epsilon)\lambda_f(x))^2} - 1 \right) ds \right| \, \left(\nabla^2 f(x)^{1/2} v_{\text{NE}} \right) \|_2 \\
&= \frac{(1+\epsilon)\lambda_f(x)}{1 - (1+\epsilon)\lambda_f(x)} \left\| \nabla^2 f(x)^{1/2} v_{\text{NE}} \right\|_2 \\
&= \frac{(1+\epsilon)\lambda_f^2(x)}{1 - (1+\epsilon)\lambda_f(x)}.
\end{aligned}$$

Bound on M_2: We have

$$\begin{aligned}
M_2 &= \left\| \int_0^1 \nabla^2 f(x)^{-1/2} \nabla^2 f(x + s v_{\text{NSK}}) \nabla^2 f(x)^{-1/2} ds \nabla^2 f(x)^{1/2} (v_{\text{NSK}} - v_{\text{NE}}) \right\|_2 \\
&\leq \left\| \int_0^1 \frac{1}{(1 - s(1+\epsilon)\lambda_f(x))^2} ds \nabla^2 f(x)^{1/2} (v_{\text{NSK}} - v_{\text{NE}}) \right\|_2 \\
&= \frac{1}{1 - (1+\epsilon)\lambda_f(x)} \left\| \nabla^2 f(x)^{1/2} (v_{\text{NSK}} - v_{\text{NE}}) \right\|_2 \\
&\overset{(i)}{\leq} \frac{1}{1 - (1+\epsilon)\lambda_f(x)} \epsilon \left\| \nabla^2 f(x)^{1/2} v_{\text{NE}} \right\|_2 \\
&= \frac{\epsilon \lambda_f(x)}{1 - (1+\epsilon)\lambda_f(x)},
\end{aligned}$$

where the inequality in step (i) follows from Lemma 6.3. □

We now use Lemma 6.6 to prove the two claims in the lemma statement.

Proof of the bound (6.18a). Recall from the theorem statement that $\eta := \frac{1}{8} \frac{1 - \frac{1}{2}(\frac{1+\epsilon}{1-\epsilon})^2 - a}{(\frac{1+\epsilon}{1-\epsilon})^3}$. By examining the roots of a polynomial in ϵ, it can be seen that $\eta \leq \frac{1-\epsilon}{1+\epsilon} \frac{1}{16}$. By applying the inequalities (6.27b), we have

$$(1+\epsilon)\lambda_f(x) \leq \frac{1+\epsilon}{1-\epsilon} \widetilde{\lambda}_f(x) \;\leq \frac{1+\epsilon}{1-\epsilon} \eta \;\leq\; \frac{1}{16} \tag{6.30}$$

whence inequality (6.27a) implies that

$$\lambda_f(x_{\text{NSK}}) \leq \frac{\frac{1}{16}\lambda_f(x) + \epsilon \lambda_f(x)}{(1 - \frac{1}{16})^2} \;\leq\; \left(\frac{16}{225} + \frac{256}{225}\epsilon \right) \lambda_f(x) \;\leq\; \frac{16}{25} \lambda_f(x). \tag{6.31}$$

Here the final inequality holds for all $\epsilon \in (0, 1/2)$. Combining the bound (6.27b) with inequality (6.31) yields

$$\widetilde{\lambda}_f(x_{\text{NSK}}) \leq (1+\epsilon)\lambda_f(x_{\text{NSK}}) \leq (1+\epsilon)\big(\frac{16}{25}\big) \, \widetilde{\lambda}_f(x) \;\leq\; \widetilde{\lambda}_f(x),$$

where the final inequality again uses the condition $\epsilon \in (0, \frac{1}{2})$. This completes the proof of the bound (6.18a).

Proof of the bound (6.18b). This inequality has been established as a consequence of proving the bound (6.31).

6.3. Proof of Theorem 4.2. Given the proof of Theorem 4.1, it remains only to prove an appropriately modified version of Lemma 6.3. It applies to the exact and sketched Newton directions $v_{\text{NE}}, v_{\text{NSK}} \in \mathbb{R}^d$ that are defined as follows

$$v_{\text{NE}} := \arg \min_{z \in \mathcal{C} - x} \Big\{ \frac{1}{2} \|\nabla^2 f(x)^{1/2} z\|_2^2 + \langle z, \, \nabla f(x) \rangle + \frac{1}{2} \langle z, \, \nabla^2 g(x) z \rangle \Big\}, \tag{6.32a}$$

$$v_{\text{NSK}} = \arg \min_{z \in \mathcal{C} - x} \underbrace{\Big\{ \frac{1}{2} \|S \nabla^2 f(x)^{1/2} z\|_2^2 + \langle z, \, \nabla f(x) \rangle + \frac{1}{2} \langle z, \, \nabla^2 g(x) z \rangle \Big\}}_{\Psi(z;S)}. \tag{6.32b}$$

Thus, the only difference is that the Hessian $\nabla^2 f(x)$ is sketched, whereas the term $\nabla^2 g(x)$ remains unsketched. Also note that since the function g is a self-concordant barrier for the set $\mathcal{C}$, we can safely omit the constraint $\mathcal{C}$ in the definitions of sketched and original Newton steps.

Lemma 6.7. *Let $S \in \mathbb{R}^{m\times n}$ be a sub-Gaussian, ROS or JL sketching matrix, and let $x \in \mathbb{R}^d$ be a (possibly random) vector independent of S. If $m \geq c_0 \max_{x\in\mathcal{C}} \frac{\mathcal{W}(\nabla^2 f(x)^{1/2}\mathcal{K})^2}{\epsilon^2}$, then*

$$\left\|\nabla^2 f(x)^{1/2}(v_{\scriptscriptstyle NSK} - v_{\scriptscriptstyle NE})\right\|_2 \leq \epsilon \left\|\nabla^2 f(x)^{1/2} v_{\scriptscriptstyle NE}\right\|_2 \tag{6.33}$$

with probability at least $1 - c_1 e^{-c_2 m \epsilon^2}$.

Proof. We follow the basic inequality argument used in the proof of Lemma 6.3. Since v_{NSK} and v_{NE} are optimal and feasible (respectively) for the sketched Newton problem (6.32b), we have $\Psi(v_{\text{NSK}}; S) \leq \Psi(v_{\text{NE}}; S)$. Defining the difference vector $\widehat{e} := v_{\text{NSK}} - v$, some algebra leads to the basic inequality

$$\begin{aligned}\frac{1}{2}\|S\nabla^2 f(x)^{1/2}\widehat{e}\|_2^2 + \frac{1}{2}\langle \widehat{e},\, \nabla^2 g(x)\widehat{e}\rangle &\leq -\langle \nabla^2 f(x)^{1/2} v_{\text{NE}},\, S^T S \nabla^2 f(x)^{1/2}\widehat{e}\rangle \\ &\quad + \langle \widehat{e},\, \big(\nabla f(x) - \nabla^2 g(x)\big) v_{\text{NE}}\rangle.\end{aligned}$$

On the other hand since v_{NE} and v_{NSK} are optimal and feasible (respectively) for the Newton step (6.32a), we have

$$\langle \nabla^2 f(x) v_{\text{NE}} + \nabla^2 g(x) v_{\text{NE}} - \nabla f(x),\, \widehat{e}\rangle \geq 0.$$

Consequently, by adding and subtracting $\langle \nabla^2 f(x) v_{\text{NE}},\, \widehat{e}\rangle$, we find that

$$\frac{1}{2}\|S\nabla^2 f(x)^{1/2}\widehat{e}\|_2^2 + \frac{1}{2}\langle v_{\text{NE}},\, \nabla^2 g(x) v_{\text{NE}}\rangle \leq \left|\langle \nabla^2 f(x)^{1/2} v_{\text{NE}},\, \big(I_n - S^T S\big)\nabla^2 f(x)^{1/2}\widehat{e}\rangle\right|. \tag{6.34}$$

We next define the matrix $\bar{H}(x)^{1/2} := \begin{bmatrix} \nabla^2 f(x)^{1/2} \\ \nabla^2 g(x)^{1/2} \end{bmatrix}$ and the augmented sketching matrix $\bar{S} := \begin{bmatrix} S & 0 \\ 0 & I_q \end{bmatrix}$ where $q = 2n$. Then we can rewrite the inequality (6.34) as follows

$$\frac{1}{2}\|\bar{S}\bar{H}(x)^{1/2}\widehat{e}\|_2^2 \leq \left|\langle \bar{H}(x)^{1/2} v_{\text{NE}},\, \big(I_q - \bar{S}^T\bar{S}\big)\bar{H}(x)^{1/2}\widehat{e}\rangle\right|.$$

Note that the modified sketching matrix $\bar{S}$ also satisfies the conditions (6.13a) and (6.13b). Consequently the remainder of the proof follows as in the proof of Lemma 6.3.
□

7. Discussion. In this paper, we introduced and analyzed the Newton sketch, a randomized approximation to the classical Newton updates. This algorithm is a natural generalization of the Iterative Hessian Sketch (IHS) updates analyzed in our earlier work [30]. The IHS applies only to constrained least-squares problems (for which the Hessian is independent of the iteration number), whereas the Newton Sketch applies to twice differentiable convex functions, minimized over a closed and convex set. We described various applications of the Newton sketch, including its use with barrier methods to solve various forms of constrained problems. For the minimization of self-concordant functions, the combination of the Newton sketch within interior point updates leads to much faster algorithms for an extensive body of convex optimization problems.

Each iteration of the Newton sketch has lower computational complexity than classical Newton's method. Moreover, ignoring logarithmic factors, it has lower overall computational complexity than first-order methods when either $n \geq d^2$, when applied in the primal form, or $d \geq n^2$, when applied in the dual form; here n and d denote the dimensions of the data matrix A. In the context of barrier methods, the parameters n and d typically correspond to the number of constraints and number of variables, respectively. In many "big data" problems, one of the dimensions is much larger than the other, in which case the Newton sketch is advantageous. Moreover, sketches based on the randomized

Hadamard transform are well-suited to in parallel environments: in this case, the sketching step can be done in $\mathcal{O}(\log m)$ time with $\mathcal{O}(nd)$ processors. This scheme significantly decreases the amount of central computation—namely, from $\mathcal{O}(m^2 d + nd \log m)$ to $\mathcal{O}(m^2 d + \log d)$.

There are a number of open problems associated with the Newton sketch. Here we focused our analysis on the cases of sub-Gaussian, randomized orthogonal system (ROS) sketches and JL embeddings. It would also be interesting to analyze sketches based on row sampling and leverage scores. Such techniques preserve the sparsity of the Hessian, and can be used in conjunction with sparse KKT system solvers. Finally, it would be interesting to explore the problem of lower bounds on the sketch dimension m. In particular, is there a threshold below which any algorithm that has access only to gradients and m-sketched Hessians must necessarily converge at a sub-linear rate, or in a way that depends on the strong convexity and smoothness parameters? Such a result would clarify whether or not the guarantees in this paper are improvable.

Acknowledgements. Both authors were partially supported by Office of Naval Research MURI grant N00014-11-1-0688, Air Force Office of Scientific Research grant AFOSR-FA9550-14-1-0016, and National Science Foundation Grants CIF-31712-23800 and DMS-1107000. In addition, MP was supported by a Microsoft Research Fellowship. We thank Fanny Yang for helpful comments on an earlier draft of this paper.

Appendix A. Gaussian widths with ℓ_1-constraints.

In this appendix, we state and prove an elementary lemma that bounds for the Gaussian width for a broad class of ℓ_1-constrained problems. In particular, given a twice-differentiable convex function ψ, a vector $c \in \mathbb{R}^d$, a radius R and a collection of d-vectors $\{a_i\}_{i=1}^n$, consider a convex program of the form

$$\min_{x \in \mathcal{C}} \Big\{ \sum_{i=1}^n \psi\big(\langle a_i,\, x\rangle\big) + \langle c,\, x\rangle \Big\}, \qquad \text{where} \quad \mathcal{C} = \{x \in \mathbb{R}^d \mid \|x\|_1 \leq R\}. \tag{A.1}$$

Lemma A.1. *Suppose that the ℓ_1-constrained program* (A.1) *has a unique optimal solution x^* such that $\|x^*\|_0 \leq s$ for some integer s. Then denoting the tangent cone at x^* by $\mathcal{K}$, then*

$$\max_{x \in \mathcal{C}} \mathcal{W}(\nabla^2 f(x)^{1/2} \mathcal{K}) \leq 6\sqrt{s \log d}\, \sqrt{\frac{\psi''_{\max}}{\psi''_{\min}}}\; \frac{\max\limits_{j=1,\ldots,d} \|A_j\|_2}{\sqrt{\gamma_s^-(A)}},$$

where

$$\psi''_{\min} = \min_{x \in \mathcal{C}} \min_{i=1,\ldots,n} \psi''(\langle a_i,\, x\rangle, y_i), \quad \textit{and} \quad \psi''_{\max} = \max_{x \in \mathcal{C}} \max_{i=1,\ldots,n} \psi''(\langle a_i,\, x\rangle, y_i).$$

Proof. It is well-known (e.g., [18, 31]) that the tangent cone of the ℓ_1-norm at any s-sparse solution is a subset of the cone $\{z \in \mathbb{R}^d \mid \|z\|_1 \leq 2\sqrt{s}\|z\|_2\}$. Using this fact, we have the following sequence of upper bounds

$$
\begin{aligned}
\mathcal{W}(\nabla^2 f(x)^{1/2}\mathcal{K}) &= \mathbb{E}_w \max_{\substack{z^T\nabla^2 f(x) z=1\,,\\ z\in\mathcal{K}}} \langle w,\, \nabla^2 f(x)^{1/2} z\rangle \\
&= \mathbb{E}_w \max_{\substack{z^T A^T\,\mathrm{diag}\left(\psi''(\langle a_i,\,x\rangle x, y_i)\right) A z=1\,,\\ z\in\mathcal{K}}} \langle w,\, \mathrm{diag}\,(\psi''(\langle a_i,\, x\rangle, y_i))^{1/2}\, Az\rangle \\
&\leq \mathbb{E}_w \max_{\substack{z^T A^T A z\leq 1/\psi''_{\min}\\ z\in\mathcal{K}}} \langle w,\, \mathrm{diag}\,(\psi''(\langle a_i,\, x\rangle, y_i))^{1/2}\, Az\rangle \\
&\leq \mathbb{E}_w \max_{\|z\|_1\leq \frac{2\sqrt{s}}{\sqrt{\gamma_s^-(A)}}\frac{1}{\sqrt{\psi''_{\min}}}} \langle w,\, \mathrm{diag}\,(\psi''(\langle a_i,\, x\rangle, y_i))^{1/2}\, Az\rangle \\
&= \frac{2\sqrt{s}}{\sqrt{\gamma_s^-(A)}}\frac{1}{\sqrt{\psi''_{\min}}}\mathbb{E}_w \,\|A^T\,\mathrm{diag}\,(\psi''(\langle a_i,\, x\rangle, y_i))^{1/2}\, w\|_\infty \\
&= \frac{2\sqrt{s}}{\sqrt{\gamma_s^-(A)}}\frac{1}{\sqrt{\psi''_{\min}}}\mathbb{E}_w \max_{j=1,\ldots,d} \Big|\underbrace{\sum_{i=1,\ldots,n} w_i A_{ij}\psi''(\langle a_i,\, x\rangle, y_i)^{1/2}}_{Q_j}\Big|.
\end{aligned}
$$

Here the random variables Q_j are zero-mean Gaussians with variance at most

$$\sum_{i=1,\ldots,n} A_{ij}^2 \psi''(\langle a_i,\, x\rangle, y_i) \leq \psi''_{\max}\|A_j\|_2^2.$$

Consequently, applying standard bounds on the suprema of Gaussian variates [21], we obtain

$$\mathbb{E}_w \max_{j=1,\ldots,d} \Big|\sum_{i=1,\ldots,n} w_i A_{ij}\psi''(\langle a_i,\, x\rangle, y_i)^{1/2}\Big| \leq 3\sqrt{\log d}\sqrt{\psi''_{\max}}\max_{j=1,\ldots,d}\|A_j\|_2.$$

When combined with the previous inequality, the claim follows. □

REFERENCES

[1] D. Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, 2003.

[2] N. Ailon and B. Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 557–563. ACM, 2006.

[3] N. Ailon and E. Liberty. Fast dimension reduction using Rademacher series on dual BCH codes. *Discrete Comput. Geom*, 42(4):615–630, 2009.

[4] P. L. Bartlett, O. Bousquet, and S. Mendelson. Local Rademacher complexities. *Annals of Statistics*, 33(4):1497–1537, 2005.

[5] A. Bordes, L. Bottou, and P. Gallinari. Sgd-qn: Careful quasi-Newton stochastic gradient descent. *Journal of Machine Learning Research*, 10:1737–1754, 2009.

[6] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, Cambridge, UK, 2004.

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

[8] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-Newton method for large-scale optimization. *arXiv preprint arXiv:1401.7020*, 2014.

[9] A. Dasgupta, R. Kumar, and T. Sarlós. A sparse Johnson-Lindenstrauss transform. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 341–350. ACM, 2010.

[10] K. R. Davidson and S. J. Szarek. Local operator theory, random matrices, and Banach spaces. In *Handbook of Banach Spaces*, volume 1, pages 317–336. Elsevier, Amsterdam, NL, 2001.

[11] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM Journal on Numerical analysis*, 19(2):400–408, 1982.

[12] R. S. Dembo and T. Steihaug. Truncated Newton algorithms for large-scale unconstrained optimization. *Mathematical Programming*, 26(2):190–212, 1983.

[13] P. Drineas, M. Magdon-Ismail, M.W. Mahoney, and D.P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

[14] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlos. Faster least squares approximation. *Numer. Math*, 117(2):219–249, 2011.

[15] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[16] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.

[17] T. Hastie and B. Efron. LARS: Least angle regression, Lasso and forward stagewise. *R package version 0.9-7*, 2007.

[18] T. Hastie, R. Tibshirani, and M. J. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations.* CRC Press, Chapman and Hall, New York, 2015.

[19] D.M. Kane and J. Nelson. Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM*, 61(1):4, 2014.

[20] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale ℓ_1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):606–617, 2007.

[21] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer-Verlag, New York, NY, 1991.

[22] H. M. Markowitz. *Portfolio Selection*. Wiley, New York, 1959.

[23] James Martens. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 735–742, 2010.

[24] P. McCullagh and J.A. Nelder. *Generalized linear models*. Monographs on statistics and applied probability 37. Chapman and Hall/CRC, New York, 1989.

[25] J. Nelson and H. L. Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 117–126. IEEE, 2013.

[26] Y. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[27] Y. Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, New York, 2004.

[28] Y. Nesterov and A. Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics, 1994.

[29] M. R. Osborne, B. Presnell, and B. A. Turlach. On the Lasso and its dual. *Journal of Computational and Graphical Statistics*, 2(9):319–337, 2000b.

[30] M. Pilanci and M. J. Wainwright. Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. Technical report, UC Berkeley, 2014. Full length version at arXiv:1411.0347.

[31] M. Pilanci and M. J. Wainwright. Randomized sketches of convex programs with sharp guarantees. *IEEE Trans. Info. Theory*, 9(61):5096–5115, September 2015.

[32] G. Pisier. Probablistic methods in the geometry of Banach spaces. In *Probability and Analysis*, volume 1206 of *Lecture Notes in Mathematics*, pages 167–241. Springer, 1989.

[33] N. N. Schraudolph, J. Yu, and S. Günter. A stochastic quasi-newton method for online convex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 436–443, 2007.

[34] Karl Sjöstrand. Matlab implementation of lasso, lars, the elastic net and spca. *Informatics and Mathematical Modelling, Technical University of Denmark (DTU)*, 2005.

[35] D.A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

[36] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

[37] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *Compressed Sensing: Theory and Applications*, 2012.

[38] S.J. Wright and J. Nocedal. *Numerical optimization*, volume 2. Springer New York, 1999.

[39] N. Yamashita and M. Fukushima. On the rate of convergence of the Levenberg-Marquardt method. In *Topics in numerical analysis*, pages 239–249. Springer, 2001.