# QUASI-NEWTON METHODS: SUPERLINEAR CONVERGENCE WITHOUT LINE SEARCHES FOR SELF-CONCORDANT FUNCTIONS

WENBO GAO*† AND DONALD GOLDFARB†

*Industrial Engineering and Operations Research, Columbia University*
*500 W 120th St., New York, NY 10027*

ABSTRACT. We consider the use of a *curvature-adaptive* step size in gradient-based iterative methods, including quasi-Newton methods, for minimizing self-concordant functions, extending an approach first proposed for Newton's method by Nesterov. This step size has a simple expression that can be computed analytically; hence, line searches are not needed. We show that using this step size in the BFGS method (and quasi-Newton methods in the Broyden convex class other than the DFP method) results in superlinear convergence for strongly convex self-concordant functions. We present numerical experiments comparing gradient descent and BFGS methods using the curvature-adaptive step size to traditional methods on deterministic logistic regression problems, and to versions of stochastic gradient descent on stochastic optimization problems.

## 1. INTRODUCTION

We are concerned in this paper with iterative optimization algorithms, which at each step, first select a *direction* $d_k$ and then determine a *step size* $t_k$. Such algorithms, which are usually referred to as *line search* algorithms, need to choose an appropriate step size $t_k$ to perform well, both in theory and in practice.

Theoretical proofs of global convergence generally assume one of the following approaches for selecting the step sizes:

(1) The step sizes are obtained from line searches.
(2) The step size is a constant, often chosen 'sufficiently small'.

Inexact line searches, and in particular those that choose steps that satisfy the Armijo-Wolfe conditions, or just the latter combined with backtracking, are usually performed and work well in practice. However, they can be costly to perform, and are often prohibitively costly for many common objective functions such as those that arise in machine learning, computer vision, and natural language processing. Moreover, in stochastic optimization algorithms, line searches based on stochastic function values and gradients, which are only estimates of the true quantities (see Section 8), can be meaningless. In contrast, constant step sizes $t_k = t$ for all $k$ require no additional computation beyond selecting $t$, but determining an appropriate constant $t$ may be difficult. The value of $t$ required in the theoretical analysis

*E-mail address*: wg2279@columbia.edu, goldfarb@columbia.edu.
*Date*: June 03, 2018.
2010 *Mathematics Subject Classification.* 90C53, 90C30.
*Corresponding author.
†Research of this author was supported in part by NSF Grant CCF-1527809.

is often too small for practical purposes, and moreover, is impossible to compute without knowledge of unknown parameters (e.g. the Lipschitz constant of $\nabla f$). A single constant step size may also be highly suboptimal, as the iterates transition between regions with different curvature.

The basic idea for a step size determined by the local curvature of the objective function $f$ was developed by Nesterov, who introduced the *damped Newton method* [17]. This idea is closely related to a well-behaved class of functions known as *self-concordant functions* [18], which we define in Section 3. When applied to a self-concordant function $f$, the damped Newton method is globally convergent and locally converges quadratically. These results were extended in recent work.

(1) Tran-Dinh et al. [25] propose a proximal framework for composite self-concordant minimization, which includes proximal damped Newton, proximal quasi-Newton, and proximal gradient descent. They establish that proximal damped Newton is globally convergent and locally quadratically convergent, and that proximal damped gradient descent is globally convergent and locally linearly convergent. However, they do not propose a proximal quasi-Newton algorithm or prove global convergence for a generic version of such an algorithm.
(2) Zhang and Xiao [26] propose a distributed method for self-concordant empirical loss functions, based on the damped Newton method, and establish its convergence.
(3) Lu [16] proposes a randomized block proximal damped Newton method for composite self-concordant minimization, and establishes its convergence.

While the damped Newton method has been extensively studied, no comparable theory exists for quasi-Newton methods in the self-concordant setting. It is well known that for convex functions, proving global convergence for the BFGS method [2, 8, 11, 24] with inexact line searches is far more challenging than proving global convergence for scaled gradient methods, and that a similar statement holds for the $Q$-superlinear convergence of the BFGS method applied to strongly convex functions compared with, for example, proving $Q$-quadratic convergence of Newton's method. With regard to $Q$-superlinear convergence, it is well known [21] that if the the largest eigenvalue of the Hessian of the objective is bounded above, and if the sum of the distances of the iterates generated by the BFGS method from the global minimizer is finite, then the BFGS method converges $Q$-superlinearly. We note that Tran-Dinh et al. [25] give a proof of this local result for their "pure"-proximal-BFGS method (i.e., one that uses a step size of 1 on every iteration and starts from a point "close" to the global minimizer), but they do not prove that this method generates iterates satisfying the required conditions. This leaves open the question of how to design a globally convergent "dampled" version of the BFGS method for self-concordant functions. In particular, we wish to avoid assuming either the Dennis-Moré condition [7] or the summability of the distances to the global minmizer, since these conditions are extremely strong, verging on being tautological, as assumptions.

In this paper we extend the theory of self-concordant minimization developed by Nesterov and Nemirovski [18] and further developed by Tran-Dinh et al. [25]. Our focus here is mainly on filling the gap in this theory for quasi-Newton methods. To simplify the presentation, we consider only quasi-Newton methods that use the BFGS update, although our results apply to all methods in the Broyden class of quasi-Newton methods other than the DFP method [6, 9]. We introduce a framework for non-composite optimization; i.e., we do not consider proximal methods as in [25]. The key feature of this framework is a step size that is optimal with respect to an upper bound on the decrease in the objective value, which we call the

*curvature-adaptive step size*. We use the term curvature-adaptive, or simply *adaptive*, to refer to this step size choice or to methods that employ it, so as not to confuse such methods with damped BFGS updating methods (e.g., see [20, §18.3]), which are unrelated.

We first prove that scaled gradient methods that use the curvature-adaptive step size are globally $R$-linearly convergent on strongly convex self-concordant functions. We note that in [25], this step size is also identified, but that the $R$-linear convergence is only proved locally. We then prove our main result, on quasi-Newton methods: that the BFGS method, using this step size, is globally convergent for functions that are self-concordant, bounded below, and have a bounded Hessian, and furthermore, is $Q$-superlinearly convergent when the function is strongly convex and self-concordant. For completeness, we then present several numerical experiments which shed insight on the behavior of adaptive methods. These show that for deterministic optimization, using curvature-based step sizes in quasi-Newton methods is dominated by using inexact line searches, whereas in stochastic settings, using curvature-based step sizes is very helpful compared to constant step sizes.

Our paper is organized as follows. In Section 2, we introduce the notation and assumptions that we use throughout the paper. In Section 3, we define the class of self-concordant functions and describe their essential properties. In Section 4, we introduce our framework for self-concordant minimization and provide a derivation of what we call the *curvature-adaptive* step size, which extends the curvature-based step size obtained in [25] for proximal gradient methods. In Section 5, we apply our approach to scaled gradient methods, and give a simple proof that these methods are globally $R$-linearly convergent on strongly convex self-concordant functions. In Section 6, we present our main results. Specifically, we prove there that the BFGS method with curvature-adaptive step sizes is globally and $Q$-superlinearly convergent. In Section 7, we present numerical experiments testing our new methods on logistic regression problems in the deterministic setting. In Section 8, we discuss stochastic extensions of adaptive methods. In Appendix A, we provide a numerical example of solving an online stochastic problem using stochastic adaptive methods.

## 2. Preliminaries

We use $f : \mathbb{R}^n \to \mathbb{R}$ to denote the objective function, and $g(\cdot), G(\cdot)$ denote the gradient $\nabla f(\cdot)$ and Hessian $\nabla^2 f(\cdot)$, respectively. In the context of a sequence of points $\{x_k\}_{k=0}^{\infty}$, we write $g_k$ for $g(x_k)$ and $G_k$ for $G(x_k)$. Unless stated otherwise, the function $f$ is assumed to have continuous third derivatives (as $f$ is generally assumed to be self-concordant), which we write as $f \in \mathcal{C}^3$.

The norm $\|\cdot\|$ denotes the 2-norm, and when applied to a matrix, the operator 2-norm.

## 3. Self-Concordant Functions

The notion of *self-concordant* functions was first introduced by Nesterov and Nemirovski [18] for their analysis of Newton's method in the context of interior-point methods. Nesterov [17] provides a clear exposition and motivates self-concordancy by observing that, while Newton's method is invariant under affine transformations, the convergence analysis makes use of norms which are *not* invariant. To remedy this, Nesterov and Nemirovski replace the Euclidean norm by an invariant local norm, and replace the assumption of Lipschitz continuity of the Hessian $G(x)$ by the self-concordancy of $f$.

**Definition.** *Let $f$ be a convex function. The* local norm *of $h \in \mathbb{R}^n$ at a point $x$ where $G(x) \succ 0$ is given by*

$$\|h\|_x = \sqrt{h^T G(x) h}.$$

**Definition.** *A closed convex function $f : \mathbb{R}^n \to \mathbb{R}$ is* self-concordant *if $f \in \mathcal{C}^3$ and there exists a constant $\kappa$ such that for every $x \in \mathbb{R}^n$ and every $h \in \mathbb{R}^n$, we have*

$$|\nabla^3 f(x)[h,h,h]| \leq \kappa (\nabla^2 f(x)[h,h])^{3/2}.$$

*If $\kappa = 2$, $f$ is* standard self-concordant*. Any self-concordant function can be scaled to be standard self-concordant; the scaled function $\frac{1}{4}\kappa^2 f$ is standard self-concordant. Hence, we assume all self-concordant functions have $\kappa = 2$, unless stated otherwise.*

There is also an equivalent definition which is frequently useful.

**Theorem 3.1** (Lemma 4.1.2, [17])**.** *A closed convex function $f$ is self-concordant if and only if for every $x \in \mathbb{R}^n$ and all $u_1, u_2, u_3 \in \mathbb{R}^n$, we have*

$$|\nabla^3 f(x)[u_1,u_2,u_3]| \leq 2 \prod_{i=1}^{3} \|u_i\|_x.$$

The next inequalities are fundamental for self-concordant functions. These results are well known (see [17, §4.1.4]), but for completeness, we provide a proof.

**Lemma 3.2.** *Let $f$ be standard self-concordant and strictly convex, and let $x \in \mathbb{R}^n$ and $0 \neq d \in \mathbb{R}^n$. Let $\delta = \|d\|_x$. Then for all $t \geq 0$,*

$$f(x+td) \geq f(x) + t g(x)^T d + \delta t - \log(1+\delta t) \tag{3.1}$$

*and*

$$g(x+td)^T d \geq g(x)^T d + \frac{\delta^2 t}{1+\delta t}. \tag{3.2}$$

*For all $0 \leq t < \frac{1}{\delta}$,*

$$f(x+td) \leq f(x) + t g(x)^T d - \delta t - \log(1-\delta t) \tag{3.3}$$

*and*

$$g(x+td)^T d \leq g(x)^T d + \frac{\delta^2 t}{1-\delta t}. \tag{3.4}$$

*Proof.* Define $\phi : \mathbb{R} \to \mathbb{R}$ by $\phi(t) = d^T \nabla^2 f(x+td) d$. Since $f$ has continuous third derivatives, $\phi(t)$ is continuously differentiable and from the definition of self-concordancy, its derivative satisfies

$$|\phi'(t)| = |\nabla^3 f(x+td)[d,d,d]| \leq 2(\nabla^2 f(x+td)[d,d])^{3/2} = 2\phi(t)^{3/2}. \tag{3.5}$$

Moreover, since $f$ is strictly convex and $d \neq 0$, $\phi(t) > 0$ for all $t$. Therefore, from (3.5),

$$\left|\frac{d}{dt}\phi(t)^{-1/2}\right| = \frac{1}{2}|\phi(t)^{-3/2}\phi'(t)| \leq 1.$$

Defining $\psi(s) = \frac{d}{dt}\phi(t)^{-1/2}\big|_{t=s}$, the above inequality is equivalent to $|\psi(s)| \leq 1$. By Taylor's Theorem, there exists a point $u \in (0,t)$ such that $\phi(t)^{-1/2} - \phi(0)^{-1/2} = t\psi(u)$. Since $|\psi(u)| \leq 1$, we deduce that

$$\phi(0)^{-1/2} - t \leq \phi(t)^{-1/2} \leq \phi(0)^{-1/2} + t.$$

Note that $\delta = \phi(0)^{1/2}$. Rearranging the upper bound, we find that for all $t \geq 0$,

$$\phi(t) \geq \frac{\delta^2}{(1+\delta t)^2}. \tag{3.6}$$

Similarly, we find that for $0 \leq t < \frac{1}{\delta}$,

$$\phi(t) \leq \frac{\delta^2}{(1-\delta t)^2}. \tag{3.7}$$

Integrating (3.6) yields the inequalities (3.1), (3.2), and integrating (3.7) produces (3.3), (3.4). □

## 4. Curvature-Adaptive Step Sizes

We define a general framework for an iterative method with step sizes determined by the local curvature. At each step, we compute a descent direction $d_k = -H_k g_k$, where $H_k$ is a positive definite matrix, and a step size

$$t_k = \frac{\rho_k}{(\rho_k + \delta_k)\delta_k},$$

where

$$\delta_k = \|d_k\|_{x_k}$$

and

$$\rho_k = g_k^T H_k g_k.$$

We then advance to the point $x_{k+1} = x_k + t_k d_k$.

We will refer to the above step size $t_k$ as the *curvature-adaptive* step size, or simply the *adaptive* step size. A method within our framework will be referred to as an *adaptive* method. A generic method in this framework is specified in Algorithm 1.

Note that this framework encompasses several classical methods. When $H_k = I$ for all $k$, the resulting method is gradient descent. When $H_k = G_k^{-1}$, we recover the *damped Newton method* proposed by Nesterov. When $H_k$ is an approximation of $G_k^{-1}$ obtained by applying a quasi-Newton updating formula, the resulting method is a quasi-Newton method. In particular, we will focus on the case where $H_k$ evolves according to the BFGS update formula. We also note that in all variants other than the damped Newton method, we do not access the full Hessian matrix $G_k$ at any step, but only the action of $G_k$ on the direction $d_k$, which typically requires a computational effort similar to that required to compute the gradient $g_k$.

Using the results of Section 3, we now show that the curvature-adaptive step size $t_k = \frac{\rho_k}{(\rho_k+\delta_k)\delta_k}$ in Algorithm 1 maximizes a lower bound on the decrease in $f$ obtained by taking a step in the direction $d_k$.

**Lemma 4.1.** *Suppose $f$ is self-concordant and strictly convex. At iteration $k$ of Algorithm 1, taking the step $t_k d_k$, where $d_k = -H_k g_k$ and $t_k = \frac{\rho_k}{(\rho_k+\delta_k)\delta_k}$, yields the point $x_{k+1} = x_k + t_k d_k$ at which the objective function $f(x_{k+1})$ satisfies*

$$f(x_{k+1}) \leq f(x_k) - \omega(\eta_k) \tag{4.1}$$

*where*

$$\eta_k = \frac{\rho_k}{\delta_k}$$

*and $\omega : \mathbb{R} \to \mathbb{R}$ is the function $\omega(z) = z - \log(1+z)$.*

**Algorithm 1** Adaptive Iterative Method

**input**: $x_0, H_0$, variant
**for** $k = 0, 1, 2, \ldots$ **do**
  Set $d_k \leftarrow -H_k g_k$
  Set $\rho_k \leftarrow -g_k^T d_k$
  Set $\delta_k^2 \leftarrow d_k^T G_k d_k$
  Set $t_k \leftarrow \frac{\rho_k}{(\rho_k+\delta_k)\delta_k}$
  Set $x_{k+1} \leftarrow x_k + t_k d_k$
  **if** variant (i): gradient descent **then**
    $H_{k+1} \leftarrow I$
  **end if**
  **if** variant (ii): Newton **then**
    $H_{k+1} = G_{k+1}^{-1}$
  **end if**
  **if** variant (iii): BFGS **then**
    Use standard BFGS formula (6.1) to compute $H_{k+1}$
  **end if**
  **if** variant (iv): L-BFGS **then**
    Update L-BFGS curvature pairs
  **end if**
**end for**

*Moreover, the step size $t_k$ minimizes the upper bound (3.3) on $f(x_{k+1})$ provided by Lemma 3.2.*

*Proof.* We fix the index $k$ and omit the subscripts for brevity. First, observe that

$$0 \leq t = \frac{\rho}{(\rho+\delta)\delta} < \frac{1}{\delta}.$$

Therefore, we can apply inequality (3.3) to $f(x+td)$. Noting that $\rho = -g^T d$, (3.3) can be written as $f(x+td) \leq f(x) - \Delta(t)$ where $\Delta(\cdot)$ is defined to be the function $\Delta(\tau) = (\rho+\delta)\tau + \log(1-\delta\tau)$. For the curvature-adaptive step size $t$, it is easily verified that

$$\Delta(t) = \Delta\left(\frac{\rho}{(\rho+\delta)\delta}\right) = \frac{\rho}{\delta} + \log\left(\frac{\delta}{\rho+\delta}\right) = \frac{\rho}{\delta} - \log\left(1+\frac{\rho}{\delta}\right) = \omega(\eta).$$

Furthermore, for $0 \leq \tau < \frac{1}{\delta}$, $\frac{d}{d\tau}\Delta(\tau) = \rho + \delta - \frac{\delta}{1-\delta\tau}$ and $\frac{d^2}{d\tau^2}\Delta(\tau) = -\frac{\delta^2}{(1-\delta\tau)^2}$. We find that $\frac{d}{d\tau}\Delta(t) = 0$ and $\frac{d^2}{d\tau^2}\Delta(t) \leq 0$, which implies that $\Delta$ is maximized at $\tau = t = \frac{\rho}{(\rho+\delta)\delta}$. □

Since $\omega(\eta) = \eta - \log(1+\eta)$ is positive for all $\eta > 0$, it follows that if $\limsup_k \eta_k > 0$, then $f(x_k) \to -\infty$. This simple fact will be crucial in our convergence analysis.

**Lemma 4.2.** *If, in addition to the assumptions in Lemma 4.1, $f$ is bounded below, then $\eta_k = \frac{\rho_k}{\delta_k} \to 0$ for any of the adaptive variants in Algorithm 1.*

*Proof.* By Lemma 4.1, $f(x_k)$ satisfies $f(x_k) \leq f(x_0) - \sum_{j=0}^{k-1} \omega(\eta_j)$. Suppose that $\limsup_k \eta_k > 0$. Since the function $\omega(\eta)$ is positive and monotonically increasing for $\eta > 0$, we have $\limsup_k \omega(\eta_k) = \omega(\limsup_k \eta_k) > 0$. Hence $f(x_k) \to -\infty$, a contradiction. □

In terms of $g_k, H_k$, and $G_k$, the adaptive step size $t_k$ can be expressed as

$$t_k = \frac{g_k^T H_k g_k}{g_k^T H_k G_k H_k g_k + g_k^T H_k g_k \sqrt{g_k^T H_k G_k H_k g_k}}.$$

This formula relates $t_k$ to the local curvature. When the curvature of $f$ in the direction $d_k = -H_k g_k$ is relatively flat, the local norm $\|d_k\|_{x_k} = \sqrt{g_k^T H_k G_k H_k g_k}$ is small, and the adaptive step size $t_k$ will be large. Conversely, when the curvature of $f$ in the direction $d_k$ is steep, $t_k$ will be small. Intuitively, this is precisely the desired behavior for a step size, since we wish to take larger steps when the function changes slowly, and smaller steps when the function changes rapidly.

## 5. Scaled Gradient Methods

We first consider the class of methods where the matrices $H_k$ are positive definite and uniformly bounded above and below. That is, there exist positive constants $\lambda, \Lambda$ such that for every $k \geq 0$,

$$\lambda I \preceq H_k \preceq \Lambda I. \tag{5.1}$$

The convergence analysis is rather straightforward, as seen in the proofs of the following two theorems for these methods.

**Theorem 5.1.** *If $f$ is self-concordant, strictly convex, bounded below, and the Hessian satisfies $G(x) \preceq MI$ on the level set $\Omega = \{x : f(x) \leq f(x_0)\}$, then any adaptive method (Algorithm 1) for which the matrices $H_k$ satisfy equation* (5.1) *converges globally in the sense that $\lim_{k\to\infty} \|g_k\| = 0$.*

*Proof.* Since $H_k$ is positive definite, $H_k^{1/2}$ exists and we may define $z_k = H_k^{1/2} g_k$. Observe that

$$\eta_k = \frac{g_k^T H_k g_k}{\sqrt{g_k^T H_k G_k H_k g_k}} = \frac{z_k^T z_k}{\sqrt{z_k^T (H_k^{1/2} G_k H_k^{1/2}) z_k}} \geq \frac{\|z_k\|}{\sqrt{\Lambda M}} \geq \sqrt{\frac{\lambda}{\Lambda M}} \|g_k\| \tag{5.2}$$

where we have used the fact that the maximum eigenvalue of $H_k^{1/2} G_k H_k^{1/2}$ is bounded by $\Lambda M$. By Lemma 4.2, $\eta_k \to 0$. Therefore $\|g_k\| \to 0$. □

If in addition, $f$ is strongly convex with $mI \preceq G(x)$ for $m > 0$, then an adaptive method satisfying equation (5.1) is globally $R$-linearly convergent. The proof uses the fact that strongly convex functions satisfy the *Polyak-Łojasiewicz inequality*, which is stated in the following well known lemma (e.g., see [21, 10]).

**Lemma 5.2.** *If $f$ is strongly convex with $mI \preceq G(x)$, and $x_*$ is the unique minimizer of $f$, then $\|g(x)\|^2 \geq 2m(f(x) - f(x_*))$.*

We are now ready to prove the $R$-linear convergence of adaptive scaled gradient methods.

**Theorem 5.3.** *If $f$ is self-concordant and strongly convex (so there exist constants $0 < m \leq M$ such that $mI \preceq G(x) \preceq MI$ for all $x \in \Omega$), then an adaptive method (Algorithm 1) for which the matrices $H_k$ satisfy equation* (5.1) *is globally $R$-linearly convergent. That is, there exists a positive constant $\gamma < 1$ such that $f(x_{k+1}) - f(x_*) \leq \gamma(f(x_k) - f(x_*))$ for all $k$.*

*Proof.* Since $\eta_k \to 0$ by Lemma 4.2, the sequence $\{\eta_k\}_{k=0}^{\infty}$ is bounded. Let $\Gamma = \sup_k \eta_k < \infty$, and let $c = \frac{1}{2(1+\Gamma)}$. Observe that $\omega(z) = z - \log(1+z) \geq cz^2$ for $0 \leq z \leq \Gamma$, as $\omega(0) = 0$ and $\frac{d}{dz}(\omega(z) - cz^2) = \frac{z(1-2c-2cz)}{1+z}$, which is non-negative for $0 \leq z \leq \Gamma$. Hence, since $\eta_k \leq \Gamma$ for all $k$, we have

$$\begin{aligned} f(x_{k+1}) - f(x_*) \leq f(x_k) - f(x_*) - \omega(\eta_k) &\leq f(x_k) - f(x_*) - c\eta_k^2 \\ &\leq f(x_k) - f(x_*) - \frac{c\lambda}{\Lambda M}\|g(x_k)\|^2 \\ &\leq \left(1 - \frac{\lambda m}{\Lambda(1+\Gamma)M}\right)(f(x_k) - f(x_*)) \end{aligned}$$

where the first line follows from inequality (3.3), the second from inequality (5.2), and the third from Lemma 5.2. Taking $\gamma = 1 - \frac{\lambda m}{\Lambda(1+\Gamma)M}$, we obtain the desired $R$-linear convergence. □

5.1. **Adaptive Gradient Descent.** When $H_k = I$ for all $k$ in Algorithm 1, the method corresponds to gradient descent with adaptive step sizes that incorporate second-order information. This strategy for selecting analytically computable step sizes may have several advantages in practice. Using second-order information allows a better local model of the objective function. The classical analysis of gradient descent with a fixed step size also generally requires a sufficiently small step size in order to guarantee convergence. This step size is a function of the Lipschitz constant for the gradient $g(x)$, which is either unknown or impractical to compute. The step size needed to ensure convergence in theory is also often impractically tiny, leading to slow convergence in practice. For the class of self-concordant functions, an adaptive step size can be easily computed without knowledge of any constants, and still provides a theoretical guarantee of convergence, which is a significant advantage.

A proximal gradient descent method with adaptive step sizes was studied by Tran-Dinh et al. [25], who proved the method to be globally convergent for self-concordant functions, and locally $R$-linearly convergent for strongly convex self-concordant functions. However, our convergence analysis above employs different techniques from those in [25], and in particular, we obtain the following theorem, which shows that the adaptive gradient descent method is globally $R$-linearly convergent, as an immediate corollary of Theorem 5.1 and Theorem 5.3:

**Theorem 5.4.** *Suppose that $f$ is self-concordant, strictly convex, bounded below, and $G(x) \preceq MI$ on the level set $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$. Then the adaptive gradient descent method converges in the sense that $\lim_{k\to\infty} \|g_k\| = 0$. Furthermore, if $f$ is strongly convex on $\Omega$, then the adaptive gradient descent method is globally $R$-linearly convergent.*

5.2. **Adaptive L-BFGS.** The limited-memory BFGS algorithm (L-BFGS, [15]) stores a fixed number of previous *curvature pairs* $(s_k, y_k)$, where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$, and computes $d_k = -H_k g_k$ from the curvature pairs using a two-loop recursion [19]. It is well known that L-BFGS satisfies equation (5.1). In [12], the following bounds are obtained.

**Theorem 5.5** (Lemma 1, [12])**.** *Suppose that $f$ is strongly convex, with $mI \preceq G(x) \preceq MI$. Let $\ell$ be the number of curvature pairs stored by the L-BFGS method. Then the matrices $H_k$ satisfy*

$$\lambda I \preceq H_k \preceq \Lambda I,$$

*where $\lambda = (1 + \ell M)^{-1}$ and $\Lambda = (1 + \sqrt{\kappa})^{2\ell}\left(1 + \frac{1}{m(2\sqrt{\kappa}+\kappa)}\right)$ for $\kappa = M/m$.*

Hence, it follows immediately from Theorem 5.1 and Theorem 5.3 that:

**Theorem 5.6.** *Suppose that $f$ is self-concordant, strongly convex, and $mI \preceq G(x) \preceq MI$ on the level set $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$. Then the adaptive L-BFGS method is globally R-linearly convergent.*

We note that, as with gradient descent, it is well known that the L-BFGS method converges if inexact Armijo-Wolfe line searches are performed, or a sufficiently small fixed step size, that depends on the Lipschitz constant of $g(x)$, is used.

## 6. Adaptive BFGS

If $H_k$ is chosen to approximate $(\nabla^2 f(x_k))^{-1}$, then we obtain quasi-Newton methods with adaptive step sizes. In particular, we may iteratively update $H_k$ using the BFGS update formula, which we briefly describe. Let $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. The BFGS update sets $H_{k+1}$ to be the nearest matrix to $H_k$ (in a variable metric) satisfying the *secant equation* $H_{k+1}y_k = s_k$[11]. It is well known that $H_{k+1}$ has the following expression in terms of $H_k, s_k$ and $y_k$:

$$H_{k+1} = \frac{s_k s_k^T}{y_k^T s_k} + \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) H_k \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right). \tag{6.1}$$

### 6.1. Superlinear Convergence of Adaptive BFGS.

The convergence analysis of the classical BFGS method [21, 4] assumes that the method uses inexact line searches satisfying the *Armijo-Wolfe* conditions: for constants $c_1, c_2 \in (0,1)$ with $c_1 < \frac{1}{2}$ and $c_1 < c_2$, the step size $t_k$ should satisfy

$$f(x_k + t_k d_k) \leq f(x_k) + c_1 t_k g_k^T d_k \qquad \text{(Armijo condition)} \tag{6.2}$$

and

$$g(x_k + t_k d_k)^T d_k \geq c_2 g_k^T d_k. \qquad \text{(Wolfe condition)} \tag{6.3}$$

Under the assumption of Armijo-Wolfe line searches, Powell [21] proves the following global convergence theorem for BFGS.

**Theorem 6.1** (Lemma 1, [21])**.** *If the BFGS algorithm with Armijo-Wolfe inexact line search is applied to a convex function $f(x)$ that is bounded below, if $x_0$ is any starting vector and $H_0$ is any positive definite matrix, and if the Hessian $G(x)$ satisfies $G(x) \preceq MI$ for all $x$ in the level set $\Omega = \{x : f(x) \leq f(x_0)\}$, then the limit*

$$\liminf_{k \to \infty} \|g_k\| = 0 \tag{6.4}$$

*is obtained.*

In our setting, $f$ is a self-concordant and strictly convex function that is bounded below and satisfies $G(x) \preceq MI$. In order to prove that adaptive BFGS is convergent in the sense of the limit (6.4), it suffices to show that the adaptive step sizes $t_k$ satisfy the Armijo condition for any $c_1 < \frac{1}{2}$, and eventually satisfy the Wolfe condition for any $c_2 < 1$ (i.e. there exists some $k_0$ such that the Wolfe condition is satisfied for all $k \geq k_0$). Specifically, we prove the following two theorems that apply to *every* adaptive method described by Algorithm 1.

**Theorem 6.2.** *Let $f$ be self-concordant and strictly convex. The curvature-adaptive step size $t_k = \frac{\rho_k}{(\rho_k + \delta_k)\delta_k}$ satisfies the Armijo condition for any $c_1 \leq \frac{1}{2}$.*

*Proof.* Let $c_1 \leq \frac{1}{2}$. We aim to prove that $f(x_{k+1}) \leq f(x_k) + c_1 t_k g_k^T d_k$. By Lemma 4.1, $f(x_{k+1}) \leq f(x_k) - \omega(\eta_k)$. Therefore, it suffices to prove that

$$\omega(\eta_k) \geq -\frac{1}{2} t_k g_k^T d_k.$$

For brevity, we omit the index $k$. Notice that

$$-tg^T d = tg^T Hg = t\rho = \frac{\rho^2}{(\rho+\delta)\delta} = \frac{\rho^2/\delta^2}{\rho/\delta + 1} = \frac{\eta^2}{1+\eta}.$$

Therefore, we must prove that for $\eta \geq 0$,

$$\omega(\eta) = \eta - \log(1+\eta) \geq \frac{1}{2}\frac{\eta^2}{1+\eta}.$$

Define $\zeta(z) = z - \log(1+z) - \frac{1}{2}\frac{z^2}{1+z}$. Observe that $\zeta(0) = 0$ and

$$\frac{d}{dz}\zeta(z) = 1 - \frac{1}{1+z} - \frac{1}{2}\frac{z^2+2z}{(1+z)^2} = \frac{1}{2}\frac{z^2}{(1+z)^2}.$$

Since $\frac{d}{dz}\zeta(z) \geq 0$ for all $z \geq 0$, we conclude that $\omega(\eta) \geq \frac{1}{2}\frac{\eta^2}{1+\eta}$ for all $\eta \geq 0$. This completes the proof. □

**Theorem 6.3.** *Let $f$ be self-concordant, strictly convex, and bounded below. Suppose that $\{x_k\}_{k=0}^{\infty}$ is a sequence of iterates generated by Algorithm 1. For any $0 < c_2 < 1$, there exists an index $k_0$ such that for all $k \geq k_0$, the curvature-adaptive step size $t_k$ satisfies the Wolfe condition.*

*Proof.* We aim to prove that $g_{k+1}^T d_k \geq c_2 g_k^T d_k$. This is equivalent to $g(x_k + t_k d_k)^T d_k - g(x_k)^T d_k \geq -(1-c_2) g(x_k)^T d_k = (1-c_2)\rho_k$. By inequality (3.2) with $\delta = \delta_k$ and $t = t_k$, we have

$$g(x_k + t_k d_k)^T d_k - g(x_k)^T d_k \geq \frac{\delta_k^2 t_k}{1 + \delta_k t_k} = \frac{\delta_k \rho_k}{2\rho_k + \delta_k} = \frac{1}{1 + 2\eta_k}\rho_k. \tag{6.5}$$

Since $f$ is bounded below, Lemma 4.2 implies that $\eta \to 0$, and hence there exists some $k_0$ such that $\frac{1}{1+2\eta_k} \geq 1 - c_2$ for all $k \geq k_0$. □

Note that the assumption of strict convexity also implies that $y_k^T s_k > 0$, so the BFGS update is well-defined.

We can now immediately apply Theorem 6.1 to deduce that adaptive BFGS is convergent. Since there always exists an index $k_0$ such that the Armijo-Wolfe conditions are satisfied for all $k \geq k_0$, we can consider the subsequent iterates $\{x_k\}_{k=k_0}^{\infty}$ as arising from the classical BFGS method with initial matrix $H_{k_0}$.

**Theorem 6.4.** *Let $f$ be self-concordant, strictly convex, bounded below, whose Hessian satisfies $G(x) \preceq MI$ for all $x \in \Omega$. Then for the adaptive BFGS method, $\liminf_{k\to\infty} \|g_k\| = 0$.*

It is also possible to directly prove Theorem 6.4 by analyzing the evolution of the trace and determinant of $H_k$, but the resulting proof, which is quite long, does not provide clarity on the essential properties of the adaptive step size.

It is well known that if the objective function $f$ is strongly convex, then the classical BFGS method converges $Q$-superlinearly. Let us now assume that $f$ is strongly convex, so

there exist constants $0 < m \leq M$ with $mI \preceq G(x) \preceq MI$ for all $x \in \Omega$. Let $x_*$ denote the unique minimizer of $f$.

**Theorem 6.5** (Lemma 4, [21])**.** *Let $f$ be strongly convex, and let $\{x_k\}_{k=0}^{\infty}$ be the sequence of iterates generated by the BFGS method with inexact Armijo-Wolfe line searches. Then $\sum_{k=0}^{\infty} \|x_k - x_*\| < \infty$.*

Since the adaptive step size $t_k$ eventually satisfies the Armijo-Wolfe conditions, the same holds for BFGS with adaptive step sizes.

**Theorem 6.6.** *Let $f$ be self-concordant and strongly convex. The sequence of iterates $\{x_k\}_{k=0}^{\infty}$ produced by adaptive BFGS satisfies $\sum_{k=0}^{\infty} \|x_k - x_*\| < \infty$.*

In the proof of superlinear convergence for the classical BFGS method, it is assumed that the Hessian $G(x)$ is Lipschitz continuous. However, it is unnecessary to make this assumption in our setting, as $G(x)$ is necessarily Lipschitz when $f$ is self-concordant and $G(x)$ is bounded above. This fact is not difficult to establish, but we provide a proof for completeness.

**Theorem 6.7.** *If $f$ is self-concordant and satisfies $G(x) \preceq MI$ for all $x \in \Omega$, then $G(x)$ is Lipschitz continuous on $\Omega$, with constant $2M^{3/2}$.*

*Proof.* Let $x, y \in \Omega$, and let $e = x - y$. Let $v \in \mathbb{R}^n$ be any unit vector. By Taylor's Theorem, we have

$$v^T G(x) v = v^T G(y) v + \int_0^1 \nabla^3 f(y + \tau e)[v, v, e] d\tau.$$

Hence, by Theorem 3.1,

$$\begin{aligned} |v^T (G(x) - G(y)) v| &\leq \int_0^1 |\nabla^3 f(y + \tau e)[v, v, e]| d\tau \\ &\leq 2 \int_0^1 v^T G(y + \tau e) v \sqrt{e^T G(y + \tau e) e} d\tau \\ &\leq 2 \int_0^1 M^{3/2} \|e\| d\tau = 2M^{3/2} \|x - y\|. \end{aligned}$$

Therefore, the eigenvalues of $G(x) - G(y)$ are bounded in norm by $2M^{3/2}\|x - y\|$. It follows that $\|G(x) - G(y)\| \leq 2M^{3/2}\|x - y\|$, so $G(x)$ is Lipschitz continuous. □

It is well known that the BFGS method is invariant under an affine change of coordinates, so we may assume without loss of generality that $G(x_*) = I$. This corresponds to considering the function $\widetilde{f}(\widetilde{x}) = f(G(x_*)^{-1/2}\widetilde{x})$ and performing a change of coordinates $\widetilde{x} = G(x_*)^{1/2} x$. By [17, Theorem 4.1.2], the function $\widetilde{f}$ is also self-concordant, with the same $\kappa$ as for $f$.

To complete the proof of superlinear convergence, we use results established by Dennis and Moré [7] and Griewank and Toint [13]. In [13, §4], Griewank and Toint prove that, given Theorem 6.6 and Lipschitz continuity of $G(x)$ (Theorem 6.7), the following limit holds:

$$\lim_{k \to \infty} \frac{\|(B_k - I) d_k\|}{\|d_k\|} = 0 \tag{6.6}$$

Furthermore, the argument in [13] shows that both $\{\|B_k\|\}_{k=0}^{\infty}$ and $\{\|H_k\|\}_{k=0}^{\infty}$ are bounded. Writing $B_k d_k = -g_k$ and $-d_k = H_k g_k$, and using the fact that $\|d_k\| \leq \|H_k\| \|g_k\| \leq \Gamma \|g_k\|$,

where $\Gamma$ is an upper bound on the sequence of norms $\{\|H_k\|\}_{k=0}^{\infty}$, we have an equivalent limit

$$\lim_{k\to\infty} \frac{\|H_k g_k - g_k\|}{\|g_k\|} = 0 \tag{6.7}$$

This enables us to prove that the adaptive step sizes $t_k$ converge to 1, which is necessary for superlinear convergence.

**Theorem 6.8.** *The curvature-adaptive step sizes $t_k$ in the adaptive BFGS method converge to 1.*

*Proof.* We omit the index $k$ for brevity, and define $u = Hg - g$. Since $t$ can be expressed as $t = \frac{\eta/\delta}{1+\eta}$, and we have from Lemma 4.2 that $\eta \to 0$, it suffices to show that $\frac{\eta}{\delta}$ converges to 1.

$$\begin{aligned}\frac{\eta}{\delta} = \frac{\rho}{\delta^2} &= \frac{g^T H g}{g^T H G H g} \\ &= \frac{g^T g + g^T u}{g^T G g + 2 g^T G u + u^T G u} \\ &= \frac{1 + \frac{g^T u}{g^T g}}{\frac{g^T G g}{g^T g} + 2\frac{g^T G u}{g^T g} + \frac{u^T G u}{g^T g}}\end{aligned}$$

The limit (6.7) implies that $\frac{\|u\|}{\|g\|} \to 0$. Hence, the Cauchy-Schwarz inequality and the upper bound $G(x) \preceq MI$ imply that $\frac{g^T u}{g^T g}, \frac{g^T G u}{g^T g}, \frac{u^T G u}{g^T g}$ converge to 0. Since $G = G(x_k)$ and $x_k \to x_*$, we have $G \to I$, and therefore $\frac{g^T G g}{g^T g} \to 1$. It follows that $\frac{\eta}{\delta}$, and therefore $t$, converges to 1. □

We now make a slight modification to the Dennis-Moré characterization of superlinear convergence. Using the triangle inequality twice and the fact that $G(x_*) = I$, we obtain

$$\begin{aligned}\frac{\|(B_k - I)s_k)\|}{\|s_k\|} &= \frac{\|t_k g_{k+1} - t_k g_k - G(x_*)s_k - t_k g_{k+1}\|}{\|s_k\|} \\ &\geq t_k \frac{\|g_{k+1}\|}{\|s_k\|} - \frac{\|t_k g_{k+1} - t_k g_k - t_k G(x_*)s_k - (1-t_k)G(x_*)s_k\|}{\|s_k\|} \\ &\geq t_k \frac{\|g_{k+1}\|}{\|s_k\|} - \frac{t_k \|\int_0^1 (G(x_k + \tau s_k) - G(x_*))s_k d\tau\|}{\|s_k\|} - |1-t_k|\frac{\|G(x_*)s_k\|}{\|s_k\|}.\end{aligned}$$

Rearranging, and applying Theorem 6.7,

$$\frac{\|g_{k+1}\|}{\|s_k\|} \leq \frac{1}{t_k}\frac{\|(B_k - I)s_k)\|}{\|s_k\|} + 2M^{3/2}\max\{\|x_k - x_*\|, \|x_{k+1} - x_*\|\} + \frac{|1-t_k|}{t_k}M. \tag{6.8}$$

Since $x_k \to x_*$ by Theorem 6.4 and $t_k \to 1$ by Theorem 6.8, both of the latter terms converge to 0. Finally, equation (6.6) implies that $\frac{\|(B_k - I)s_k\|}{\|s_k\|}$ converges to 0, so it follows from equation (6.8) that $\frac{\|g_{k+1}\|}{\|s_k\|}$ converges to 0.

Since $f$ is strongly convex, $\|g(x)\| \geq m\|x - x_*\|$. Hence, we find that

$$\frac{\|g_{k+1}\|}{\|s_k\|} \geq \frac{m\|x_{k+1} - x_*\|}{\|x_{k+1} - x_*\| + \|x_k - x_*\|},$$

which implies that $\frac{\|x_{k+1}-x_*\|}{\|x_k-x_*\|} \to 0$. Thus, we have the following:

**Theorem 6.9.** *Suppose that $f$ is self-concordant, and strongly convex on $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$. Then the adaptive BFGS method converges Q-superlinearly.*

By the same reasoning, the results in [4] and [13] imply that these convergence theorems also hold for the adaptive versions of the quasi-Newton methods in *Broyden's convex class*, with the exception of the DFP method. The adaptive versions of the Block BFGS methods proposed in [10] can also be shown to be $Q$-superlinearly convergent.

6.2. **Hybrid Step Selection.** Consider the damped Newton method of Nesterov, which is obtained by setting $H_k = G_k^{-1}$. This yields $\rho_k = g_k^T G_k^{-1} g_k$ and $\delta_k = \sqrt{g_k G_k^{-1} G_k G_k^{-1} g_k} = \sqrt{\rho}$, whence $\eta = \rho/\delta = \delta$. The curvature-adaptive step size $t$ then reduces to

$$t = \frac{\eta/\delta}{1+\eta} = \frac{1}{1+\delta}.$$

When $\delta$ is large (for example, if the initial point $x_0$ is chosen poorly), then the curvature-adaptive step size may be very small, even when the inverse Hessian approximation $H_k$ is good. This conservatism is the price of the curvature-adaptive step size guaranteeing global convergence (in contrast to Newton's method, which is *not* globally convergent, and to gradient descent, which may diverge if the step size is too large). A small step $t_k d_k$ is likely to result in $t_{k+1}$ also being small[1]. Thus, when the initial $\delta$ is large, a method using adaptive step sizes may produce a long succession of small steps. This suggests the following heuristic for selecting step sizes:

(1) Select a set $T_k$ of candidate step sizes for $t_k$.
(2) At step $k$, test the elements of $T_k$ in order until a candidate step size is found which satisfies the Armijo condition (6.2).
(3) If no element of $T_k$ satisfies the Armijo condition, then set $t_k$ to be the adaptive step size.

For instance, in our numerical experiments reported in Section 7, we take $T_k$ to be $(1, \frac{1}{4}, \frac{1}{16})$ for all $k$. This allows the method to take steps of size $t_k = 1$ when 1 satisfies the Armijo condition, which is desirable for reducing the number of iterations needed before superlinear convergence kicks in.

We refer to this scheme as *hybrid step selection*. For a proper choice of $T_k$, hybrid step selection avoids the disadvantage of exclusively using adaptive step sizes, where the step size may be small for many iterations. It will also generally be more efficient to compute than a full line search, since no more than $|T_k|$ candidate step sizes are tested before switching to the adaptive step size.

## 7. Numerical Experiments

To compare our adaptive methods to classical algorithms, we solve several binary classification problems using *logistic regression*. In these problems, the objective function to be

[1]As an illustrative example, consider applying the damped Newton method to the quadratic function $f(x) = \frac{1}{2}\|x\|^2$. Since $d_k = -x_k$ and $\delta_k = \|x_k\|$, we have $t_k = \frac{1}{1+\|x_k\|}$ and $x_{k+1} = \frac{\|x_k\|}{1+\|x_k\|} x_k$. If $\|x_0\|$ is large, then it is clear that the damped Newton method will take many tiny steps until $\|x_k\|$ is sufficiently reduced. This is in stark contrast to Newton's method, which reaches the minimizer after a single step.

minimized has the form

$$L(w) = \frac{1}{N}\sum_{i=1}^{N} \log(1 + \exp(-y_i x_i^T w)) + \frac{1}{2N}\|w\|_2^2. \tag{7.1}$$

where the training data $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ consists of feature vectors $x_i \in \mathbb{R}^n$ and classifications $y_i \in \{-1, +1\}$. Zhang and Xiao [26] showed that the logistic regression objective function $L(w)$ is self-concordant.

**Theorem 7.1** (Lemma 1, [26])**.** *Let $B = \max_i \|x_i\|$. The scaled function $\frac{B^2 N}{4} L(w)$ is standard self-concordant.*

In our tests, we compared seven algorithms:

(1) BFGS with adaptive step sizes (BFGS-A).
(2) BFGS with Armijo-Wolfe line search (BFGS-LS).
(3) BFGS with hybrid step selection (BFGS-H), using $T_k = (1, \frac{1}{4}, \frac{1}{16})$.
(4) L-BFGS with adaptive step sizes (LBFGS-A), using the past $\ell = \min\{\frac{n}{2}, 20\}$ curvature pairs.
(5) L-BFGS with Armijo-Wolfe line search (LBFGS-LS), using the past $\ell = \min\{\frac{n}{2}, 20\}$ curvature pairs.
(6) Gradient descent with adaptive step sizes (GD-A).
(7) Gradient descent with Armijo-Wolfe line search (GD-LS).

An initial Hessian approximation $H_0$ must be provided for the BFGS and L-BFGS methods. It is easy, but not necessarily effective, to simply take $H_0 = I$. Another common strategy for initializing $H_0$, described in [20], that is often quite effective, is to take $H_0 = I$ on the first step, and then, before performing the first BFGS update (6.1), scale $H_0$:

$$H_0 \leftarrow \frac{y_0^T s_0}{y_0^T y_0} I. \tag{7.2}$$

It is easy to verify that the scaling factor $y_0^T s_0 / y_0^T y_0$ lies between the smallest and largest eigenvalues of the inverse of the average Hessian $\overline{G} = \int_0^1 G(x_0 + \tau s_0) d\tau$ along the initial step.

Similarly, for the L-BFGS method, the initial matrix used at step $k+1$ in the two-loop recursion is chosen as:

$$H_0 \leftarrow \frac{y_k^T s_k}{y_k^T y_k} I.$$

We refer to this as *identity scaling*.

The line search used the `WolfeLineSearch` routine from the minFunc software package [23]. The Armijo-Wolfe parameters were $c_1 = 0.1, c_2 = 0.75$, and the line search was configured to use an initial step size $t = 1$ and perform quadratic interpolation (`LS_interp = 1`, `LS_multi = 0`).

We chose six data sets from LIBSVM [5] with a variety of dimensions, which are listed in Table 1. We plot the progress of each algorithm as a function of CPU time used. The progress is measured by the *log gap* $\log_{10}(f(w) - f(w_*))$, where $w_*$ is a pre-computed optimal solution. The starting point was always set to $w_0 = 0$. All algorithms were terminated when either the gradient reached the threshold $\|g(x)\| < 10^{-7}$, or after 480 seconds of CPU time. A brief summary of the results can be found in Table 2, which lists the number of iterations taken by the BFGS-type methods for convergence.

| Data set | $n$ | $N$ |
|---|---|---|
| `covtype.libsvm.binary.scale` | 55 | 581012 |
| `ijcnn1.tr` | 23 | 35000 |
| `leu` | 7130 | 38 |
| `rcv1_train.binary` | 47237 | 20242 |
| `real-sim` | 20959 | 72309 |
| `w8a` | 301 | 49749 |

TABLE 1. Data sets used in Section 7

| Data set | $n$ | Identity Scaling | Number of iterations | | |
|---|---|---|---|---|---|
| | | | BFGS-A | BFGS-LS | BFGS-H |
| `covtype.libsvm.binary.scale` | 55 | No | 844 | 80 | 126 |
| | | Yes | 1532 | 458 | 479 |
| `ijcnn1.tr` | 23 | No | 286 | 36 | 66 |
| | | Yes | 434 | 142 | 162 |
| `w8a` | 301 | No | 2254 | 240 | 637 |
| | | Yes | 2506 | 398 | 653 |
| `leu` | 7130 | No | 1197 | 95 | 293 |
| | | Yes | 909 | 177 | 251 |
| `rcv1_train.binary` | 47237 | No | 161 | 31 | 35 |
| | | Yes | 284 | 217 | 232 |
| `real-sim` | 20959 | No | 356 | 44 | 55 |
| | | Yes | 592 | 247 | 317 |

TABLE 2. The number of iterations until convergence of the BFGS methods.

Our algorithms were implemented in Matlab 2017a and run on an Intel i5-6200U processor. While the CPU time is clearly platform-dependent, we sought to minimize implementation differences between the algorithms to make the test results as comparable as possible.

In Figure 1, we plot the results for the data sets `covtype.libsvm.binary.scale`, `ijcnn1.tr`, and `w8a`. On these problems, we implemented BFGS with a dense Hessian; that is, the matrices $H_k$ were stored explicitly and updated using the formula (6.1). In Table 2, we list the number of iterations used by the BFGS-type methods.

In Figure 2, we plot the results for the data sets `leu`, `rcv1_train.binary`, and `real-sim`. These problems had a large number of variables ($n > 7000$), which made it infeasible to store $H_k$ explicitly. On these problems, BFGS was implemented using the two-loop recursion with unlimited memory, and $H_0$ was kept fixed throughout the iteration process. If the number of iterations exceeds roughly $n/4$, then this approach would in fact require more memory than storing $H_k$ explicitly. However, this never occurred in our tests, as shown in Table 2.

In our tests, we found that BFGS-A required more time than BFGS-LS. Although the cost of a single step was initially lower for BFGS-A than BFGS-LS, BFGS-A often took numerous small steps in succession, making very slow progress. This situation was exactly our motivation for devising the hybrid step selection described in Section 6.2, and unfortunately, appears to occur often. However, BFGS-H achieved comparable speed to that of BFGS-LS with $T = (1, \frac{1}{4}, \frac{1}{16})$, which suggests that always trying $t = 1$ first is an excellent

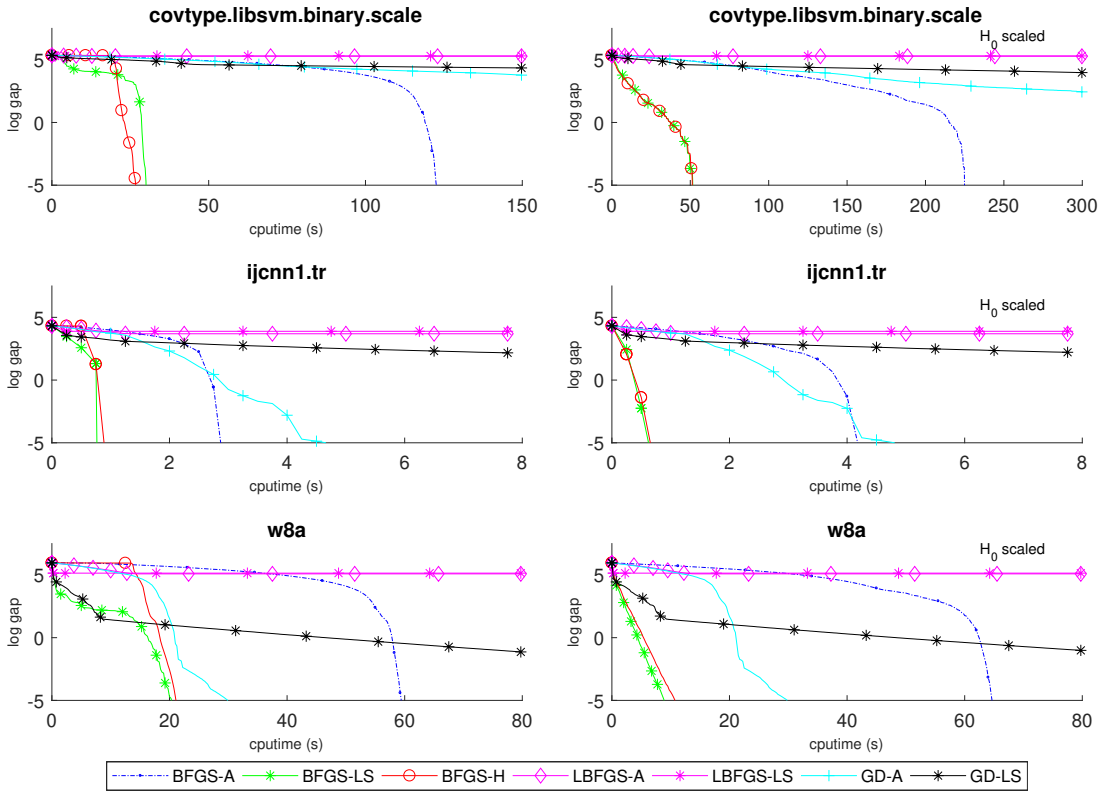


FIGURE 1. Experiments on problems with small $n$. The log gap is defined as $\log_{10}(f(w) - f(w_*))$. The loss functions are scaled to be standard self-concordant. All BFGS and L-BFGS plots on the left take $H_0 = I$, and those on the right use identity scaling.

heuristic. These results also provide evidence of the effectiveness of performing inexact line searches, in settings where it is practical to do so. In Table 3, the number of steps needed until we consistently have $t_k \approx 1$ is shown.

Since computing $t_k$ also requires a Hessian-vector product, the cost comparison between the adaptive step size and inexact line search reverses when the algorithm nears convergence. Initially, a Hessian-vector product is faster than performing multiple backtracking iterations and repeatedly testing for the Armijo-Wolfe conditions; however, the line search (and the hybrid step selection) will eventually accept the step size $t_k = 1$ immediately, becoming essentially free, whereas computing the adaptive step size continues to require a Hessian-vector product on every step.

Curiously, L-BFGS was far more effective on the problems with large $n$ (Figure 2) than on those with small $n$ (Figure 1). Both LBFGS-A and LBFGS-LS were ineffective on the problems with small $n$, which suggests that the problem lies with the step directions computed by L-BFGS, rather than the step sizes. Identity scaling was also beneficial for L-BFGS on problems with large $n$, substantially reducing the convergence time in some cases. We note that we did not experiment comprehensively with varying $\ell$, the number of curvature pairs stored in L-BFGS, and used a standard choice of $\ell = \min\{\frac{n}{2}, 20\}$. Other choices of $\ell$ might lead to very different results on the problems in our test set.

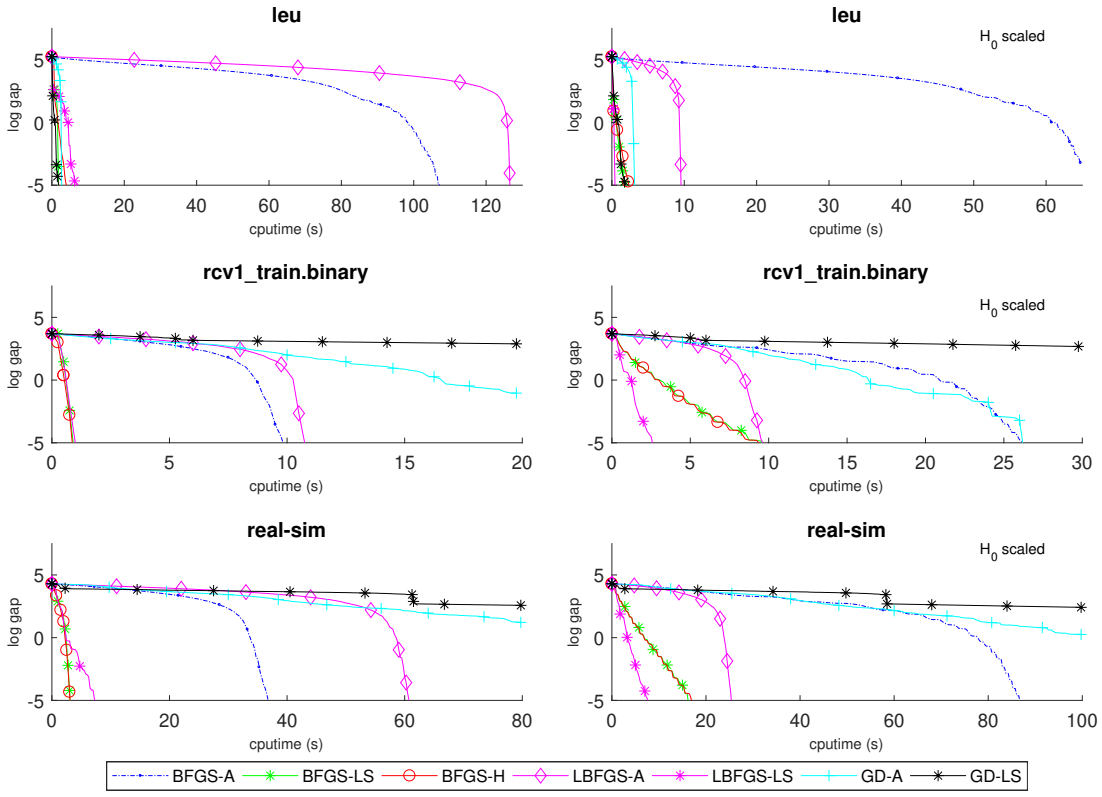


FIGURE 2. Experiments on problems with large $n$. The log gap is defined as $\log_{10}(f(w) - f(w_*))$. The loss functions are scaled to be standard self-concordant. All BFGS and L-BFGS plots on the left take $H_0 = I$, and those on the right use identity scaling.

| Data set | $n$ | Identity Scaling | Number of iterations | | |
|---|---|---|---|---|---|
| | | | BFGS-A | BFGS-LS | BFGS-H |
| `covtype.libsvm.binary.scale` | 55 | No | 797 | 57 | 62 |
| | | Yes | 1378 | 2 | 2 |
| `ijcnn1.tr` | 23 | No | 270 | 25 | 26 |
| | | Yes | 369 | 3 | 2 |
| `w8a` | 301 | No | 2056 | - | 289 |
| | | Yes | 2250 | 5 | 2 |
| `leu` | 7130 | No | - | - | 42 |
| | | Yes | 818 | 2 | 2 |
| `rcv1_train.binary` | 47237 | No | 132 | 15 | 4 |
| | | Yes | 205 | 3 | 4 |
| `real-sim` | 20959 | No | 294 | 18 | 17 |
| | | Yes | 490 | 2 | 2 |

TABLE 3. The number of iterations until $t_k = 1$ was consistently accepted by BFGS-LS and BFGS-H, and, for BFGS-A, the number of iterations until $t_k \geq 0.9$ for at least 80% of the remaining iterations. A dash '-' indicates that the condition was not met before the stopping criterion was satisfied.

On the other hand, identity scaling appeared to be detrimental for the BFGS-type methods on most problems, which can be seen from the plots in Figure 1 and Figure 2 by comparing the CPU time needed for convergence. For instance, on the data set `covtype.libsvm.binary.scale`, the time to convergence for BFGS-A increased from 120s to 225s, and from 25s to 50s for BFGS-LS and BFGS-H. In fact, identity scaling was beneficial for the BFGS-A method *only* on the data set `leu`. The data set `leu` appears to be quite different from the other problems tested. The number of training samples for `leu` was $m = 38$, while for all other problems, $m$ was at least 20,000. Moreover, gradient descent with Armijo-Wolfe line search (GD-LS) was among the fastest methods on `leu`, while on the other test problems it was significantly outperformed by BFGS. The iteration counts shown in Table 2 and Table 3 also indicate that identity scaling worsened the performance of the BFGS methods on every problem except `leu`. Curiously, performing identity scaling led to BFGS-H accepting $t_k = 1$ at a much earlier iteration on all problems, yet the total CPU time used by BFGS-H was longer for `covtype.libsvm.binary.scale`, `rcv1.train.binary`, and `real-sim`.

GD-A was surprisingly effective, outperforming GD-LS on every problem except for `leu`. This is somewhat surprising (in light of the performance of BFGS-A and BFGS-LS), and suggests that the curvature-adaptive step size may be useful for selecting hyperparameters for first-order methods.

## 8. Application to Stochastic Optimization

The adaptive step size can readily be extended to *stochastic* optimization methods. Consider a problem of the form

$$L(w) = \frac{1}{N}\sum_{i=1}^{N} f_i(w) + h(w). \tag{8.1}$$

If $N$ is extremely large, as is often the case in machine learning, simply evaluating $L(w)$ is an expensive operation, and line search is entirely impractical. To solve problems of the form (8.1), stochastic algorithms such as Stochastic Gradient Descent (SGD, [1]) select a random subset $S_k$ of $\{f_1, \ldots, f_N\}$ at step $k$ and compute the gradient for the subsampled problem

$$L^{(S_k)}(w) = \frac{1}{|S_k|}\sum_{f_i \in S_k} f_i(w) + h(w) \tag{8.2}$$

as an approximation to the gradient of the loss function (8.1), and take a step using an empirically determined small and decreasing step size. In variance-reduced versions of SGD such as SVRG [14], it is common to use a constant step size, determined through experimentation. The curvature-adaptive step size has two desirable properties in this setting: it eliminates the need to select a step size through ad-hoc experimentation, and it incorporates second-order information, which is currently not exploited by most stochastic algorithms.

Since the time of the initial writing of this article, related work on stochastic quasi-Newton methods has appeared in the machine learning literature. In particular, the curvature-adaptive step size was extended to stochastic gradient descent and stochastic BFGS in [27]. A complete discussion of stochastic optimization, and the content of [27], is beyond the scope of this article. However, we have performed several preliminary experiments with stochastic versions of adaptive methods, which are presented in Appendix A, along with a

summary of the relevant theory from [27]. These experiments (see Figure 3) demonstrate that stochastic adaptive BFGS can be quite effective for solving stochastic problems.

There is currently also little work on algorithms exploiting the finite sum structure (8.1), which can provably attain superlinear convergence. Aside from [27], we are only aware of the Newton Incremental Method (NIM) of Rodomanov and Kropotov [22], and the DiSCO method of Zhang and Xiao [26], both of which are based on Newton's method. These methods require additional memory of the order $O(N)$, which is often substantial. We are hopeful that use of the adaptive step size, and the principles behind it, will lead to new advances in the field of stochastic optimization.

## Acknowledgements

We would like to thank Jorge Nocedal for carefully reading and providing very helpful suggestions for improving an earlier version of this paper. We also thank several anonymous referees for their comments and suggestions.

## References

[1] L. Bottou, *Large-scale machine learning with stochastic gradient descent*, in Proceedings of COMPSTAT'2010, Physica-Verlag HD, pp. 177–186.
[2] C. G. Broyden, *Quasi-Newton methods and their application to function minimisation*, Mathematics of Computation, 21 (1967), pp. 368–381.
[3] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, *A stochastic quasi-newton method for large-scale optimization*, SIAM Journal on Optimization, 26 (2016), pp. 1008–1031.
[4] R. H. Byrd, J. Nocedal, and Y.-X. Yuan, *Global convergence of a class of quasi-Newton methods on convex problems*, Siam. J. Numer. Anal., (1987), pp. 1171–1190.
[5] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM Trans. Intell. Syst. Technol., 2 (2011).
[6] W. C. Davidon, *Variable metric method for minimization*, Tech. Rep. ANL-5990, Argonne National Laboratory, November 1959.
[7] J. E. Dennis Jr. and J. J. Moré, *Characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28 (1974), pp. 549–560.
[8] R. Fletcher, *A new approach to variable metric algorithms*, The Computer Journal, 13 (1970), pp. 317–322.
[9] R. Fletcher and M. J. D. Powell, *A rapidly convergent descent method for minimization*, The Computer Journal, 6 (1963), pp. 163–168.
[10] W. Gao and D. Goldfarb, *Block BFGS methods*, SIAM Journal on Optimization, 28 (2018), pp. 1205–1231.
[11] D. Goldfarb, *A family of variable-metric methods derived by variational means*, Mathematics of Computation, 24 (1970), pp. 23–26.
[12] R. Gower, D. Goldfarb, and P. Richtárik, *Stochastic block BFGS : Squeezing more curvature out of data*, in JMLR: Workshop and Conference Proceedings, vol. 48, 2016.
[13] A. Griewank and P. L. Toint, *Local convergence analysis for partitioned quasi-Newton updates*, Numer. Math., 39 (1982), pp. 429–448.
[14] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in Advances in Neural Information Processing Systems, 2013, pp. 315–323.
[15] D. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Math. Prog., 45 (1989), pp. 503–528.
[16] Z. Lu, *Randomized block proximal damped Newton methods for composite self-concordant minimization*, in review. arXiv:1607.00101, (2016).
[17] Yu. Nesterov, *Introductory Lectures on Convex Optimization*, Springer Science+Business Media, New York, 2004.
[18] Yu. Nesterov and A. Nemirovski, *Interior-point polynomial algorithms in convex programming*, Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[19] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Math. Comp., 35 (1980), pp. 773–782.
[20] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Science+Business Media, New York, 2nd ed., 2006.
[21] M. J. D. Powell, *Some global convergence properties of a variable metric algorithm for minimization without exact line searches*, in Nonlinear Programming, R. Cottle and C. Lemke, eds., vol. IX, SIAM-AMS Proceedings, 1976.
[22] A. Rodomanov and D. Kropotov, *A superlinearly-convergent proximal Newton-type method for the optimization of finite sums*, in Proceedings of the 33rd International Conference on Machine Learning, 2016, pp. 2597–2605.
[23] M. Schmidt, *minFunc: unconstrained differentiable multivariate optimization in MATLAB*, `http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html`, (2005).
[24] D. F. Shanno, *Conditioning of quasi-Newton methods for function minimization*, Mathematics of computation, 24 (1970), pp. 647–656.
[25] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher, *Composite self-concordant minimization*, Journal of Machine Learning Research, 16 (2015), pp. 371–416.
[26] Y. Zhang and L. Xiao, *DiSCO: Distributed optimization for self-concordant empirical loss*, in JMLR: Workshop and Conference Proceedings, vol. 32, 2015, pp. 362–370.
[27] C. Zhou, W. Gao, and D. Goldfarb, *Stochastic adaptive quasi-Newton methods for minimizing expected values*, Proceedings of the 34th ICML (PMLR), 70 (2017), pp. 4150–4159.

## Appendix A. Stochastic Experiments

The experiments presented here are derived from the experiments in [27, §4]. Several stochastic algorithms are tested on an *online least-squares* problem of the form

$$\min_w \mathbb{E}(Y - X^T w)^2 + \frac{1}{2}\lambda\|w\|^2.$$

*Online* refers to the method of sampling: we can only access $(X, Y)$ by calling an oracle at each iteration $k$, which returns $|S_k|$ i.i.d instances of $(X, Y)$. The model for $(X, Y)$ has the following specification:

- $X$ has a multivariate normal distribution $N(0, \Sigma)$, where $\Sigma$ is the covariance matrix of the `w8a` data set (see Table 1).
- $Y = X^T\beta + \epsilon$, where $\beta$ is deterministic and sparse (80% sparsity) and $\epsilon \sim N(0, 1)$ is a noise component.

Since our model is based on the `w8a` data set, the dimension of $w$ is $p = 300$, and the regularizer is set to $\lambda = \frac{1}{p}$.
We compare the following algorithms. For a deterministic method $M$, the corresponding *stochastic $M$ method* takes the step of the underlying $M$ method, but computed from the empirical objective function (8.2) sampled at each iteration. The convergence of these methods[2] is analyzed in [27]. In summary, the stochastic adaptive gradient descent method returns an $\epsilon$-optimal solution in expectation after $O(\log(\epsilon^{-1}))$ iterations when $|S_k|$ is chosen as a constant (depending on $\epsilon$), and stochastic adaptive BFGS converges $R$-superlinearly with probability 1 when $|S_k|$ increases superlinearly.

**SBFGS-A:** The stochastic adaptive BFGS method. At each iteration, an adaptive BFGS step is computed from the empirical objective function (8.2). The BFGS update is computed from the pair $(d_k, G_k d_k)$ which is more stable than using the pair $(s_k, y_k)$ with the difference $y_k$ of sampled gradients (see [3, 12]).

[2]Note: the stochastic adaptive BFGS analyzed in [27] is slightly different, as it incorporates an additional Wolfe condition test.

| | Value |
|---|---|
| $\alpha_1$ | $\frac{1}{140{,}000} \approx$ `7.14e-6` |
| $\alpha_2$ | `5e-6` |
| $\alpha_3$ | `2e-6` |
| $\alpha_4$ | `1e-6` |

TABLE 4. Constant step sizes.

**SBFGS-1:** The stochastic BFGS method with *constant* step size $\alpha_1$. The step size $\alpha_1$ is given in Table 4.
**SN-A:** Nesterov's stochastic damped Newton method [17].
**SN-1:** The stochastic Newton method with constant step size $\alpha_1$.
**SGD-A:** Stochastic adaptive gradient descent.
**SGD-*i*:** Stochastic gradient descent with constant step size $\alpha_i$ for $i = 1, \ldots, 4$ (Table 4).

The theory [27] suggests taking an increasing number of samples for stochastic adaptive methods. For SBFGS-A, SBFGS-1, SN-A, SN-1, and SGD-A, we use $|S_k| = \frac{1}{2}p \cdot (1.05)^{\lfloor \frac{k}{50} \rfloor}$. That is, the number of samples starts at $\frac{1}{2}p = 150$ and increases by 5% every 50 iterations. For SGD-$i$ methods, we test three different constant batch sizes: a *small* batch of $|S_k| = \frac{1}{2}p$, a *medium* batch of $|S_k| = p$, and a *large* batch of $|S_k| = 4p$.
The results of the experiments are shown in Figure 3. As before, the *log gap* is $\log_{10}(f(x_k) - f(x_*))$, where $x_*$ is the true minimizer ($x_*$ can be computed explicitly given $\Sigma$ and $\beta$). The plots in the first column shows the trajectory of each method in 60 seconds of CPU time; the second and third columns show the final 10 seconds (from 50s to 60s) in greater detail. The starting point in all trials was $w = 0$.
Both SBFGS-A and SN-A exhibit superlinear convergence once they approach the minimizer. Curiously, SBFGS-A attains greater accuracy than SN-A using the same sample sizes (see second column of Figure 3); we suspect that the noisiness of sampling $G_k$ damages SN-A. These methods greatly outperform SGD, even with well-tuned step sizes. We note that SGD is quite sensitive to the choice of step size. A constant step size cannot be made much larger than $\alpha_1$; using $\frac{1}{130{,}000} \approx$ `7.69e-6` causes SGD (even with large batches) to immediately diverge. In fact, we can check that the largest eigenvalue of $\Sigma$ is approximately `1.32e5`. Furthermore, the superior performance of SBFGS-A and SN-A depends at least partially on the curvature-adaptive step size. The methods SBFGS-1 and SN-1, which use the constant step size $\alpha_1$, converge extremely slowly[3], so the success of SBFGS-A and SN-A is not solely due to the second-order information in $H_k$.
SGD-A was slower than SGD with tuned step sizes. We found that the initial adaptive step size was `1e-8`, which explains the relatively slow convergence of SGD-A. It is also worth noting that even with a small initial sample, SGD-A never produced an overly large step size causing it to diverge or oscillate, something which is not strictly guaranteed by the theory.

[3]It is possible to use even larger step sizes with these methods. We observed that stochastic BFGS and stochastic Newton can tolerate much larger constant step sizes than $\alpha_1$ without diverging wildly as SGD does. However, for stochastic BFGS, the performance is not better, and is usually much worse than using $\alpha_1$, as the algorithm escapes to a worse region before beginning to decrease slowly.

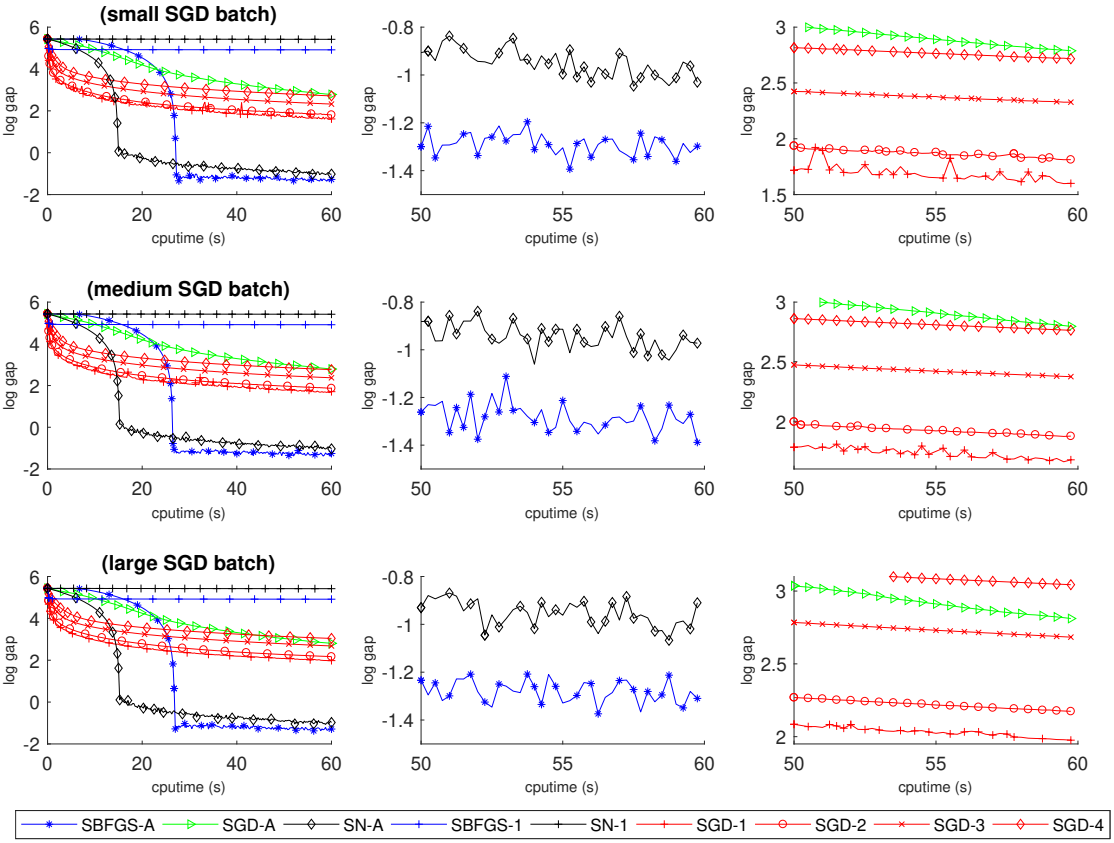


FIGURE 3. Performance of the stochastic algorithms. In the top row, the SGD methods SGD-1, SGD-2, SGD-3, SGD-4 use small batches ($|S_k| = \frac{1}{2}p$). Likewise, the second and third row use medium and large batches, respectively. The first column shows the performance of each method in 60s of CPU time, and the second and third columns show a close-up of the last 10s (50s-60s).

We have not touched on the subject of variance reduction, which is generally crucial, though not particularly relevant when considering the results in Figure 3. Good variance reduction techniques will be important for designing an effective, general-purpose solver based on SBFGS-A, SN-A, or indeed, any other of the stochastic algorithms tested.