A Self-Correcting Variable-Metric Algorithm for Stochastic Optimization

Frank E. Curtis FRANK.E.CURTIS@GMAIL.COM
Department of ISE, Lehigh University, 200 W. Packer Ave., Bethlehem, PA 18015 USA

Abstract

An algorithm for stochastic (convex or nonconvex) optimization is presented. The algorithm is variable-metric in the sense that, in each iteration, the step is computed through the product of a symmetric positive definite scaling matrix and a stochastic (mini-batch) gradient of the objective function, where the sequence of scaling matrices is updated dynamically by the algorithm. A key feature of the algorithm is that it does not overly restrict the manner in which the scaling matrices are updated. Rather, the algorithm exploits fundamental *self-correcting properties* of BFGS-type updating—properties that have been overlooked in other attempts to devise quasi-Newton methods for stochastic optimization. Numerical experiments illustrate that the method and a limited memory variant of it are stable and outperform (mini-batch) stochastic gradient and other quasi-Newton methods when employed to solve a few machine learning problems.

1. Introduction

Practical gradient-based algorithms for minimizing a smooth *deterministic* objective function can be characterized as falling between two extremes. At one extreme are steepest descent methods, i.e., algorithms in which each step is computed as a scalar times a negative gradient of the objective function. Such methods represent the extreme of incurring relatively cheap per-iteration costs while only being able to guarantee a linear rate of convergence. At the other extreme are Newton methods in which each step is computed by minimizing a local second-order Taylor series model of the objective function. Such methods incur the extra costs of having to compute second-order derivatives and solve linear systems of equations of dimension equal to the number of variables, but at the benefit of being able to guarantee a quadratic rate of convergence to a strong local minimizer. More detailed information can be found, e.g., in (Nocedal & Wright, 2006).

These extremes are magnified when one considers algorithms for solving smooth *stochastic* optimization problems, where one finds an even wider expanse between stochastic gradient (SG) (Robbins & Monro, 1951) and batch-Newton methods. For various reasons, such as its superior asymptotic large-scale learning trade-off (Bottou & Bousquet, 2008; Bottou, 2010), many algorithms for solving optimization problems for large-scale machine learning (ML) have historically been based on SG technology. However, this baseline promises to change. After all, for a variety of applications, no extreme offers the most computationally efficient form of algorithm. Classical SG approaches are relatively easy to implement and often perform well in practice after sufficient tuning, but recent advances have illustrated various benefits of moving beyond classical SG techniques. One such avenue has been the investigation of noise reduction methods, such as those in (Johnson & Zhang, 2013) and (Defazio et al., 2014), to name a couple. Another has been the investigation of second-order-type methods, the focus of this paper.

Such a shift away from simple gradient techniques occurred in the literature for large-scale, smooth, deterministic optimization with the advent of *variable-metric* algorithms in the 1960s, by far one of the most important developments in that field over the past few decades. This class of methods, which includes *quasi-Newton* methods such as those of the widely successful and celebrated Broyden-Fletcher-Goldfarb-Shanno (BFGS) variety (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970), often offers a satisfactory compromise between extremes. Such methods often perform well in practice, which many attribute to their strong theoretical guarantees. In particular, a marvelous feature of BFGS updating when solving smooth deterministic optimization problems is its ability to ensure a superlinear rate of convergence with only first-order derivative information and without the need for any linear system solves (Dennis & Moré, 1974).

A variety of attempts have been made to carry BFGS-type methods from the deterministic to the stochastic regime; e.g., one finds *online (limited memory) BFGS* (oBFGS and

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

oLBFGS) (Schraudolph et al., 2007), *stochastic gradient descent quasi-Newton* (SGD-QN) (Bordes et al., 2009), *regularized stochastic BFGS* (RES) (Mokhtari & Ribeiro, 2014), *stochastic quasi-Newton* (SQN) (Byrd et al., 2015), and *stochastic damped BFGS* (SDBFGS) (Wang et al., 2015). However, issues arise for these methods, especially when one aims to solve nonconvex problems, say for the training of deep neural networks (DNNs). The methods oBFGS, oLBFGS, and SGD-QN avoid differencing noisy gradient estimates by ensuring that the same random seed is used for both terms in the gradient difference. This requires two stochastic gradient estimates per iteration (rather than only one, as in SG) and relies on the objective function to be convex in order to ensure that only positive definite scaling matrices will be produced. The methods RES and SDBFGS employ regularization to ensure that the Hessian approximations are sufficiently positive definite (even for nonconvex problems in the case of SDBFGS), but this may lead to over-regularization that does not respect the true curvature of the objective function. As for SQN, it requires sub-sampled Hessian approximations, which requires some exact second-order information that must be handled carefully when the objective function is nonconvex. Each of these issues for each of these methods can be overcome to some extent, but there remains room for improvement in the design of effective quasi-Newton algorithms for stochastic optimization problems.

1.1. Contributions

This paper presents a new stochastic quasi-Newton method. The approach is based on a new strategy for devising such methods for ML, as explained in the following bullets.

- Previously proposed stochastic quasi-Newton methods have focused on the choice of *secant equation* to employ when only noisy gradient estimates are available. However, too much emphasis has been placed on this equation. After all, not all quasi-Newton methods for deterministic optimization possess the same features, despite the fact that they are all based on the classical secant equation. The most successful quasi-Newton scheme for deterministic optimization has been BFGS. As is known in the optimization literature, an explanation for its superior performance is that BFGS-type updating results in critical *self-correcting properties* not possessed by all quasi-Newton schemes; e.g., they are not possessed by the classical Davidon-Fletcher-Powell (DFP) scheme (Davidon, 1991; Fletcher & Powell, 1963; Byrd et al., 1987). A signifying feature of the algorithm proposed in this paper is that it has explicitly been designed to maintain these properties in stochastic settings.
- The key effects of the attained self-correcting properties of the proposed method are two fold. First, these properties ensure that the resulting Hessian and inverse Hessian approximation matrices remain sufficiently positive definite and bounded, which are key for ensuring theoretical convergence guarantees. Second, as demonstrated in the numerical results in this paper, these properties have a *stabilizing* effect on the method that allow it to perform well even when small batch sizes are used to compute the gradient estimates.
- In order to handle nonconvex problems, the traditional approaches in the quasi-Newton literature have been *skipping* and *damping* (Powell, 1978), with practical experience typically resulting in a preference for the latter. However, some damping approaches (such as those in RES and SDBFGS) might ruin the self-correcting properties of BFGS-type updating. The method proposed in this paper involves a damping procedure designed to maintain these properties.
- The method proposed in this paper is supported by a motivating convergence theory based on relatively loose assumptions about the objective function, which in particular is allowed to be nonconvex. In turn, this theory motivates a novel procedure that may be used to refine the stochastic gradient computation when one finds a lack of *consistency* between a subsequent search direction and an independent stochastic gradient estimate. When employed, this procedure leads to even more stability in the iteration, and suggests general conditions for adaptive gradient noise reduction.
- A limited memory variant of the proposed method is readily devised. The numerical experiments in this paper illustrate that this approach quickly yields lower training and testing losses while having a per-iteration cost that can be made comparable to that of SG.

2. Fundamentals

The problem of interest is to minimize a continuously differentiable objective $f : \mathbb{R}^d \to \mathbb{R}$ defined by the expectation, in terms of the distribution of a random variable ξ with domain Ξ, of a stochastic function $F : \mathbb{R}^d \times \Xi \to \mathbb{R}$:

$$\min_{w \in \mathbb{R}^d} f(w), \ \text{ where } \ f(w) := \mathbb{E}[F(w,\xi)]. \qquad (1)$$

Given an initial point $w_1 \in \mathbb{R}^d$, the proposed algorithm is iterative in that, for all $k \in \mathbb{N} := \{1,2,\dots\}$, it computes a sequence of iterates by the recursion

$$\begin{aligned} s_k &\leftarrow -\alpha_k M_k g_k \qquad (2) \\ \text{then } \ w_{k+1} &\leftarrow w_k + s_k, \end{aligned}$$

where $\alpha_k \in \mathbb{R}_{++}$ is a scalar stepsize, $M_k \in \mathbb{R}^{d\times d}$ is a symmetric positive definite scaling matrix, and $g_k \in \mathbb{R}^d$ is a stochastic gradient for f at w_k. A fundamental aspect of the algorithm is that the sequence $\{M_k\}$ is also updated recursively; specifically, during iteration $k \in \mathbb{N}$, the algorithm chooses $v_k \in \mathbb{R}^d$ such that $s_k^T v_k > 0$, then sets

$$M_{k+1} \leftarrow \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right)^T M_k \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right) + \frac{s_k s_k^T}{s_k^T v_k}. \quad (3)$$

As is well known (Nocedal & Wright, 2006), applying the Sherman-Morrison-Woodbury formula to (3) yields the following formula for $\{H_k\}$ where $H_k = M_k^{-1}$ for all $k \in \mathbb{N}$:

$$H_{k+1} \leftarrow \left(I - \frac{s_k s_k^T H_k}{s_k^T H_k s_k}\right)^T H_k \left(I - \frac{s_k s_k^T H_k}{s_k^T H_k s_k}\right) + \frac{v_k v_k^T}{s_k^T v_k}. \quad (4)$$

As is typical for a quasi-Newton method, $\{H_k\}$ and $\{M_k\}$ are refered to as sequences of Hessian and inverse Hessian approximations, respectively. If one were to choose

$$v_k \leftarrow y_k := g_{k+1} - g_k \quad (5)$$

(assuming $s_k^T y_k > 0$), then one obtains a standard BFGS update, but instead the proposed algorithm chooses

$$v_k \leftarrow \beta_k s_k + (1 - \beta_k)\alpha_k y_k, \quad (6)$$

where β_k is the smallest value in the interval $[0, 1]$ such that crucial bounds (introduced in §3) are satisfied.

The formula (6) is unique among stochastic quasi-Newton methods and is central in allowing the method to maintain the self-correcting properties of BFGS-type updating (see §3). For one thing, it differs from traditional damping formulas (e.g., see (Powell, 1978)) as it involves the step s_k as opposed to the product $H_k s_k$. In addition, it differs in the presence of the weight α_k applied to the stochastic gradient displacement. This choice has been made due to the fact that, if allowed, the choice $v_k \leftarrow \alpha_k y_k$ normalizes the appearance of α_k in the definition of s_k, namely (2). (Empirical evidence has demonstrated that this has a stabilizing effect.) Most importantly, however, is the choice of β_k, which, as shown in the next section, is what leads to the attainment of self-correcting properties of the method.

3. Self-Correcting Properties of BFGS

It is well known that the update (3) corresponds to a combination of a projection and a correction of the corresponding Hessian approximation. However, as these updates build upon one another from one iteration to the next, it is important to characterize properties of the resulting matrices and their effects on the computed steps after a *sequence* of updates have been performed. The purpose of this section is to show that as long as v_k is chosen to satisfy two critical inequalities (see (9)), then despite the successive projections that completely replace curvature information along $\text{span}(s_k)$ with each update, the corrections will be sufficient to ensure that the sequence of inverse Hessian approximations satisfy inequalities (see (11)) that are useful for ensuring global convergence guarantees.

Early work on convergence properties of quasi-Newton methods by Powell (1976) and others (Ritter, 1979; 1981; Werner, 1978) involved analyses that bound the growth of the traces and the determinants of $\{H_k\}$. The subsequent discussion follows (Byrd & Nocedal, 1989), which involves a simplified approach in which one bounds the growth of a combination of these quantities.

Given $H \succ 0$, consider $\gamma : \mathbb{R}^{d \times d} \to \mathbb{R}$ defined by

$$\gamma(H) = \text{trace}(H) - \ln(\det(H)).$$

It can be shown that $\gamma(H)$ is positive (in fact, at least d) and represents a measure of closeness between H and the identity matrix I (for which $\gamma(I) = d$); in particular, $\gamma(H)$ is an upper bound for the natural logarithm of the condition number of H. In addition, (4) implies that, for all $k \in \mathbb{N}$,

$$\text{trace}(H_{k+1}) = \text{trace}(H_k) - \frac{\|H_k s_k\|_2^2}{s_k^T H_k s_k} + \frac{\|v_k\|_2^2}{s_k^T v_k}$$
$$\text{and} \quad \det(H_{k+1}) = \det(H_k)\left(\frac{s_k^T v_k}{s_k^T H_k s_k}\right), \quad (7)$$

(Pearson, 1969) with which one can relate $\gamma(H_{k+1})$ and $\gamma(H_k)$. Specifically, assuming that $H_k \succ 0$ and the iterate displacement is nonzero, i.e., $s_k \neq 0$, then by defining

$$\cos\delta_k := \frac{s_k^T H_k s_k}{\|s_k\|_2 \|H_k s_k\|_2} \quad \text{and} \quad \epsilon_k := \frac{s_k^T H_k s_k}{\|s_k\|_2^2},$$

it follows from (7) that

$$\gamma(H_{k+1}) = \gamma(H_k) + \frac{\|v_k\|_2^2}{s_k^T v_k} - 1 - \ln\left(\frac{s_k^T v_k}{\|s_k\|_2^2}\right)$$
$$+ \underbrace{\ln(\cos^2\delta_k)}_{\leq 0} + \underbrace{\left(1 - \frac{\epsilon_k}{\cos^2\delta_k} + \ln\left(\frac{\epsilon_k}{\cos^2\delta_k}\right)\right)}_{\leq 0}. \quad (8)$$

Nonpositivity of the latter two terms follows since $s_k \neq 0$ and $H_k \succ 0$ imply that $\cos^2\delta_k \in (0, 1]$ and $\epsilon_k > 0$ for all $k \in \mathbb{N}$, since $\ln(\zeta) \leq 0$ for all $\zeta \in (0, 1]$, and since $1 - \zeta + \ln(\zeta) \leq 0$ for all $\zeta > 0$.

By bounding the growth of γ over $\{H_k\}$ and determining that there must exist certain iterations in which the latter terms in (8) are not too negative, one can prove the following theorem showing *self-correcting properties* of (4); e.g., see Thm. 2.1 in (Byrd & Nocedal, 1989).

Theorem 1. *Let the sequence of Hessian approximations $\{H_k\}$ be defined by (4), and suppose that, for all $k \in \mathbb{N}$, there exist $\eta \in (0, 1)$ and $\theta \in (1, \infty)$ such that*

$$\eta \leq \frac{s_k^T v_k}{\|s_k\|_2^2} \quad \textit{and} \quad \frac{\|v_k\|_2^2}{s_k^T v_k} \leq \theta. \quad (9)$$

Then, for any $p \in (0, 1)$, there exist constants $\{\iota, \kappa, \lambda\} \subset$

$\mathbb{R}_{++}$ *such that, for any* $K \in \{2, 3, \dots\}$*, the following relations hold for at least* $\lceil pK \rceil$ *values of* $k \in \{1, \dots, K\}$*:*

$$\iota \leq \frac{s_k^T H_k s_k}{\|s_k\|_2 \|H_k s_k\|_2} \text{ and } \kappa \leq \frac{\|H_k s_k\|_2}{\|s_k\|_2} \leq \lambda. \quad (10)$$

This theorem leads to the following corollary that provides useful bounds about the inverse Hessian approximations.

Corollary 2. *Let the sequence of inverse Hessian approximations* $\{M_k\}$ *be defined by* (3) *and suppose that the conditions of Theorem 1 hold. Then, for any* $p \in (0, 1)$*, there exist constants* $\{\mu, \nu\} \subset \mathbb{R}_{++}$ *such that, for any* $K \in \{2, 3, \dots\}$*, the following relations hold for at least* $\lceil pK \rceil$ *values of* $k \in \{1, \dots, K\}$*:*

$$\mu \|g_k\|_2^2 \leq g_k^T M_k g_k \text{ and } \|M_k g_k\|_2^2 \leq \nu \|g_k\|_2^2. \quad (11)$$

Theorem 1 and Corollary 2 show that as long as (9) holds for all $k \in \mathbb{N}$, then, for any $p \in (0, 1)$, the BFGS updates (3)–(4) ensure that for a fraction p of iterates the bounds (10)–(11) will hold for sufficiently small/large constants. Since p can be chosen as close to 1 as desired, the results show that for *any* number of iterations, these bounds hold for *all* iterations for sufficiently small/large constants. It is these bounds that are referred to as the *self-correcting properties* of the updating scheme. With stochastic gradient estimates, the choice (5) for v_k does not always ensure (9). However, these bounds are guaranteed to hold with (6) for sufficiently large $\beta_k \in [0, 1]$; e.g., they hold with $\beta_k = 1$.

4. Proposed Algorithm

A preliminary instance of the proposed algorithm is stated as Algorithm SC-BFGS. For simplicity, the algorithm is written with full (dense) inverse Hessian approximations. This is reasonable when d is not prohibitively large. Note, however, that a limited memory variant of the approach can readily be devised. This and other potential practical enhancements are the subject of §4.2.

4.1. Global Convergence

Global convergence and rate guarantees for Algorithm SC-BFGS can be proved under the following assumption. For the assumption, one should have in mind that, given w_k, one generates $g_k \approx \nabla f(w_k)$ according to an independently generated random seed ξ_k taking values in a set Ξ that does not depend on $\{w_j\}_{j=1}^{k-1}$. In what follows, $\mathbb{E}_{\xi_k}[\cdot]$ denotes expectation taken with respect to the distribution of ξ_k while $\mathbb{E}[\cdot]$ denotes total expectation taken with respect to all random quantities in a run of the algorithm. It is worthwhile to note that the assumption should not be taken lightly since each stochastic gradient influences the value of the corresponding inverse Hessian approximation, meaning that the quantities in each product of the sequence $\{M_k g_k\}$ are not independent. This complication is addressed further in §4.2. (For the moment, it is worthwhile to observe the similarity between (11) and (13)–(14).)

Algorithm SC-BFGS : Self-Correcting BFGS

```
Choose $w_1 \in \mathbb{R}^d$.
Set $g_1 \approx \nabla f(w_1)$.
Choose a symmetric positive definite $M_1 \in \mathbb{R}^{d \times d}$.
Choose a positive scalar sequence $\{\alpha_k\}$.
for $k = 1, 2, \dots$ do
    Set $s_k \leftarrow -\alpha_k M_k g_k$.
    Set $w_{k+1} \leftarrow w_k + s_k$.
    Set $g_{k+1} \approx \nabla f(w_{k+1})$.
    Set $y_k \leftarrow g_{k+1} - g_k$.
    With $v_k(\beta) := \beta s_k + (1 - \beta)\alpha_k y_k$, set

        $\beta_k \leftarrow \min\{\beta \in [0, 1] : v_k(\beta) \text{ satisfies } (9)\}$.

    Set $M_{k+1}$ by the formula (3) with $v_k := v_k(\beta_k)$.
end for
```

Assumption 3. *The objective function* $f : \mathbb{R}^d \to \mathbb{R}$ *is continuously differentiable and the sequence of iterates* $\{w_k\}$ *generated by Algorithm SC-BFGS is contained in an open set* $\mathcal{W} \subseteq \mathbb{R}^d$ *over which* f *is bounded below by* $f_{\min} \in \mathbb{R}$ *and* $\nabla f : \mathbb{R}^d \to \mathbb{R}^d$ *is Lipschitz continuous with Lipschitz constant* $L > 0$*, i.e., for all* $\{w, \overline{w}\} \subset \mathcal{W}$*,*

$$\|\nabla f(w) - \nabla f(\overline{w})\|_2 \leq L \|w - \overline{w}\|_2. \quad (12)$$

In addition, there exists a scalar $\rho \in \mathbb{R}_{++}$ *such that*

$$\rho \|\nabla f(w_k)\|_2^2 \leq \nabla f(w_k)^T \mathbb{E}_{\xi_k}[M_k g_k] \quad (13)$$

and there exist scalars $\sigma \in \mathbb{R}_{++}$ *and* $\tau \in \mathbb{R}_{++}$ *such that*

$$\mathbb{E}_{\xi_k}[\|M_k g_k\|_2^2] \leq \sigma + \tau \|\nabla f(w_k)\|_2^2 \quad (14)$$

for each iteration index $k \in \mathbb{N}$*.*

The following lemma highlights a critical upper bound—due to the Lipschitz bound (12)—on the expected decrease in the objective function attained in each iteration, as well as the key roles played by the bounds (13) and (14).

Lemma 4. *Under Assumption 3, the sequence of iterates* $\{w_k\}$ *generated by Algorithm SC-BFGS satisfies the following upper bounds for all* $k \in \mathbb{N}$*:*

$$\begin{aligned} &\mathbb{E}_{\xi_k}[f(w_{k+1})] - f(w_k) \\ &\leq -\alpha_k \nabla f(w_k)^T \mathbb{E}_{\xi_k}[M_k g_k] + \tfrac{1}{2}\alpha_k^2 L \mathbb{E}_{\xi_k}[\|M_k g_k\|_2^2] \\ &\leq -\alpha_k(\rho - \tfrac{1}{2}\alpha_k L \tau)\|\nabla f(w_k)\|_2^2 + \tfrac{1}{2}\alpha_k^2 L \sigma. \end{aligned}$$

Using this critical lemma, global convergence and rate guarantees for Algorithm SC-BFGS can be stated as Theorems 5 and 6 below. Similar results are known to hold for SG methods; e.g., the proof of the first theorem mimics the results leading to eq. (5.16) in (Bottou, 1998). See also (Bottou et al., 2016). The results relate to convergence *in expectation*; similar *almost sure* convergence results could also be proved using standard martingale techniques.

Theorem 5. *Under Assumption 3, suppose that the sequence of iterates $\{w_k\}$ is generated by Algorithm SC-BFGS with a stepsize sequence satisfying*

$$\sum_{k=1}^{\infty} \alpha_k = \infty \ \textit{and} \ \sum_{k=1}^{\infty} \alpha_k^2 < \infty. \tag{15}$$

Then, $\{w_k\}$ converges to a stationary point for (1) in the sense that the sequence of expected objective function values $\{\mathbb{E}[f(w_k)]\}$ converges to a finite limit and

$$\sum_{k=1}^{\infty} \alpha_k \mathbb{E}[\|\nabla f(w_k)\|_2^2] < \infty, \tag{16}$$

which, in particular, implies that

$$\liminf_{k\to\infty} \mathbb{E}[\|\nabla f(w_k)\|_2] = 0. \tag{17}$$

The next result addresses the minimization of a strongly convex objective, which is guaranteed to have a unique minimizer, call it $w_* \in \mathbb{R}^d$. Two example stepsize sequences are considered, though results similar to the latter can also be proved for related types of $\mathcal{O}(\frac{1}{k})$ sequences. Similar results have been proved for SG and other stochastic quasi-Newton methods; e.g., see (Byrd et al., 2015).

Theorem 6. *Suppose that Assumption 3 holds, the objective f is c-strongly convex, and the sequence of iterates $\{w_k\}$ is generated by Algorithm SC-BFGS.*

(a) If $\alpha_k = \alpha$ for some $\alpha \in (0, \min\{\frac{\rho}{L\tau}, \frac{1}{c\rho}\})$ for all $k \in \mathbb{N}$, then the expected optimality gap satisfies the following for all $k \in \mathbb{N}$:

$$\begin{aligned}&\mathbb{E}[f(w_k) - f(w_*)] \\ \le\ & \frac{\alpha L\sigma}{2c\rho} + (1-\alpha c\rho)^k \left(f(w_1) - f(w_*) - \frac{\alpha L\sigma}{2c\rho}\right) \\ \to\ & \frac{\alpha L\sigma}{2c\rho} \ \ \textit{(as } k \to \infty\textit{).}\end{aligned}$$

(b) If $\alpha_k = \frac{\alpha}{k}$ for some $\alpha \in (\frac{1}{c\rho}, \infty)$ for all $k \in \mathbb{N}$, then the expected optimality gap satisfies the following for all $k \in \mathbb{N}$ with $k \ge k(\alpha) := \alpha L\tau/\rho$:

$$\mathbb{E}[f(w_k) - f(w_*)] \le \frac{C(\alpha)}{k}, \tag{18}$$

where the scalar $C(\alpha) \in \mathbb{R}_{++}$ is defined as

$$\max\left\{\frac{\alpha^2 L\sigma}{2(\alpha c\rho - 1)}, k(\alpha)(f(w_{k(\alpha)}) - f(w_*))\right\}.$$

4.2. Practical Considerations

An important practical consideration related to the global convergence theory provided in the previous subsection is the fact that, without exact gradient evaluations, the inequalities (13) and (14) cannot be verified. In addition, one cannot sidestep the issue merely by ensuring that $\mathbb{E}_{\xi_k}[g_k] = \nabla f(w_k)$ for all $k \in \mathbb{N}$ since each g_k influences the choice of M_k, meaning that these quantities are not independent. On the other hand, it is comforting to note that Corollary 2 implies that if $g_k = \nabla f(w_k)$ for all $k \in \mathbb{N}$, then for any finite iterate sequence there exist constants $\{\mu, \nu\} \subset \mathbb{R}_{++}$ such that (13) and (14) hold with $\rho = \mu$, $\sigma = 0$, and $\tau = \nu$. Hence, while its conditions are not easily verified, Theorem 5 provides a foundation upon which one can expect good performance from Algorithm SC-BFGS.

Still, one might also consider variants of Algorithm SC-BFGS that attempt to approximately verify the key inequalities (13) and (14) when stochastic (mini-batch) gradients are employed. One such algorithmic variant is obtained by replacing Steps 8–11 in Algorithm SC-BFGS with the subroutine stated as Algorithm SC-BFGS-sub below. Within a user-defined computational budget, the subroutine repeatedly generates stochastic gradient estimates until (13) and (14) are satisfied with an auxiliary stochastic gradient estimate acting as a surrogate for the true gradient.

Algorithm SC-BFGS-sub : Subroutine for Algorithm SC-BFGS (replacing Steps 8–11)

```
Choose $\hat{k}_{\max} \in \mathbb{N}$ and $\{\rho, \sigma, \tau\} \subset \mathbb{R}_{++}$.
Set $\hat{k} \leftarrow 1$.
loop
    Compute independent stochastic (mini-batch) gradients $g_{k+1} \approx \nabla f(w_{k+1})$ and $\hat{g}_{k+1} \approx \nabla f(w_{k+1})$.
    Set $y_k \leftarrow g_{k+1} - g_k$.
    With $v_k(\beta) := \beta s_k + (1-\beta)\alpha_k y_k$, set
        $\beta_k \leftarrow \min\{\beta \in [0,1] : v_k(\beta) \text{ satisfies } (9)\}$.
    Set $M_{k+1}$ by the formula (3) with $v_k := v_k(\beta_k)$.
    break if
        $\rho\|\hat{g}_{k+1}\|_2^2 \le \hat{g}_{k+1}^T M_{k+1} g_{k+1}$
        and $\|M_{k+1} g_{k+1}\|_2^2 \le \sigma + \tau\|\hat{g}_{k+1}\|_2^2$.
    Set $\hat{k} \leftarrow \hat{k} + 1$.
    break if $\hat{k} > \hat{k}_{\max}$.
end loop
(Optional) if $\hat{k} > \hat{k}_{\max}$ then set $M_{k+1} \leftarrow M_k$.
```

Algorithm SC-BFGS-sub requires at least two gradient estimates per iteration (similar to oBFGS, oLBFGS, and SGD-QN), and potentially many more. Thus, to be competitive, the budget should be restricted (via $\hat{k}_{\max}$) and one should consider the optional strategy of resetting (effectively skipping, as in Step 11) the update if this budget is exceeded without finding estimates satisfying the conditions in Step 7 of the subroutine. However, it should be noted that instead of computing new stochastic gradient estimates in each iteration of Algorithm SC-BFGS-sub, one could instead take the average of estimates that may have already been computed at w_{k+1} (as was done for the experiments in §5). In this manner, the subroutine suggests general conditions to be used in a dynamic noise reduction strategy in the context of stochastic quasi-Newton methods.

A second important practical consideration is the fact that Algorithm SC-BFGS, as it is stated, employs full (dense) inverse Hessian approximations, which may be prohibitive if d is large. For such contexts, a limited memory variant of Algorithm SC-BFGS (with or without Algorithm SC-BFGS-sub) is readily devised. In such a method, call it SC-L-BFGS, rather than update $\{M_k\}$ explicitly, one stores $\{s_j\}_{j=k-m+1}^{k}$ and $\{v_j\}_{j=k-m+1}^{k}$ for some history length $m \in \mathbb{N}$, and can construct products with limited-memory inverse Hessian approximations through a two-loop recursion with a cost of $\mathcal{O}(d)$ (Nocedal, 1980). In this manner, the per-iteration costs of SC-L-BFGS (with or without Algorithm SC-BFGS-sub) can be made comparable to SG and any of the stochastic limited memory quasi-Newton approaches cited in §1. For example, following the analysis in (Byrd et al., 2015), the relative cost between an L-BFGS and an SG iteration can be as small as $1 + 2m/b$ where b is the batch size. With typical values for m, this value is very close to 1 even for a modest batch size.

5. Numerical Experiments

Algorithms SC-BFGS and SC-L-BFGS were implemented in Matlab, each with and without the subroutine stated in Algorithm SC-BFGS-sub. For brevity, the implementations of Algorithm SC-BFGS are respectively referred to as `SC` and `SC-s` (with `s` meaning *stable*) while the implementations of SC-L-BFGS are respectively referred to as `SC-L` and `SC-L-s`. For comparison, implementations of `SG`, `oBFGS`, and `oLBFGS` were also written in Matlab.

In this section, the results of experiments with a few ML test problems are described, each with the objective

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(w; x_i, y_i), \tag{19}$$

where $\{(x_i, y_i)\}_{i=1}^{n}$ represents *training* data and ℓ is a (not necessarily convex) loss function. For each test problem, algorithms were run with various input parameters and the results provided are for the inputs that yielded the lowest final *testing* loss, i.e., (19) evaluated with data from a *testing* set, as opposed to the *training* set used in the optimization. The algorithms were terminated after $\max\{n, 6400\}$ accesses of a pair (x_i, y_i) from the *training* set. For all algorithms, diminishing stepsize sequences of the form

$$\alpha_k = \omega_0/(\omega_1 + k) \ \text{ for all } \ k \in \mathbb{N} \tag{20}$$

were tested for all combinations of $\omega_0 \in \{2^0, 2^2, 2^4\}$ and $\omega_1 \in \{2^0, 2^2, 2^4\}$, and sequences of fixed stepsizes, i.e.,

$$\alpha_k = \omega_2 \ \text{ for all } \ k \in \mathbb{N}, \tag{21}$$

were tested for $\omega_2 \in \{2^{-4}, 2^{-2}, 2^0, 2^2, 2^4\}$. For all `SC*` algorithms, all combinations of $\eta \in \{2^{-2}, 2^{-4}, 2^{-6}\}$ and $\theta \in \{2^0, 2^2\}$ were tested. For `SC*-s`, all combinations of $\rho \in \{2^{-2}, 2^{-1}\} \cdot \eta$ and $\tau \in \{2^1, 2^2\} \cdot \theta$ were tested, though the choices $\hat{k}_{max} = 2$ and $\sigma = 0$ were fixed. For `oBFGS` and `oLBFGS`, following (Schraudolph et al., 2007), the stochastic gradient displacement vectors were computed for the sample $\mathcal{S}_k$ in iteration $k \in \mathbb{N}$ as

$$\frac{1}{|\mathcal{S}_k|}\sum_{i \in \mathcal{S}_k} (\nabla\ell(w_{k+1}; x_i, y_i) - \nabla\ell(w_k; x_i, y_i)) + \omega_3 s_k.$$

The values $\omega_3 \in \{2^{-6}, 2^{-4}, 2^{-2}, 2^0\}$ were tested. For all limited memory methods, $m = 5$ was used. All stochastic gradient estimates were computed by randomly selecting 64 samples uniformly from the training set.

All experiments were run using Matlab R2014b on a Macbook Air with a 1.7 GHz Intel Core i7 processor and 8GB of RAM. For each problem, all algorithms were run from the same randomly chosen starting point. Code for running `SC`, `SC-s`, `SC-L`, and `SC-L-s` is publicly available.[1]

5.1. Logistic Loss, Full (Dense) Matrices

As a first test, consider the training/testing data for `a1a` from the LIBSVM website[2] with (19) using a logistic loss function. This convex problem has $d = 123$ and $n = 1605$. Given this problem's relatively small dimension d, the performances of `SG`, `SC`, `SC-s`, and `oBFGS` are considered.

Considering stepsizes defined by (20), the lowest testing losses were found with inputs $(\omega_0, \omega_1) = (16, 1)$ for `SG`, $(\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (16, 16, \frac{1}{4}, 4, \frac{1}{8}, 8)$ for `SC`, $(\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (16, 1, \frac{1}{4}, 4, \frac{1}{8}, 16)$ for `SC-s`, and $(\omega_0, \omega_1, \omega_3) = (16, 16, \frac{1}{16})$ for `oBFGS`. Table 1 and Figure 1 illustrate the training and testing losses achieved. Since `SC-s` and `oBFGS` compute at least two gradient estimates per iteration, they performed fewer iterations. `SC` and `SC-s` achieved lower training and testing losses within the computational budget. (In Figure 1, dashed lines indicate the losses by `SC-s` and `oBFGS` at termination.)

Considering now the same methods with stepsizes defined by (21), the lowest testing losses were found with inputs $\omega_2 = 1$ for `SG`, $(\omega_2, \eta, \theta, \rho, \tau) = (4, \frac{1}{64}, 4, \frac{1}{128}, 8)$

[1] http://coral.ise.lehigh.edu/frankecurtis/software/

[2] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

Table 1. Training loss, testing loss, and CPU time (in seconds) for a1a for SG, SC, SC-s, and oBFGS with diminishing stepsizes.

Method	Training	Testing	CPU time (s)
SG	0.4305	0.4398	0.3405
SC	0.3588	0.3832	0.5389
SC-s	0.3614	0.3879	0.3705
oBFGS	0.3853	0.4096	0.4191

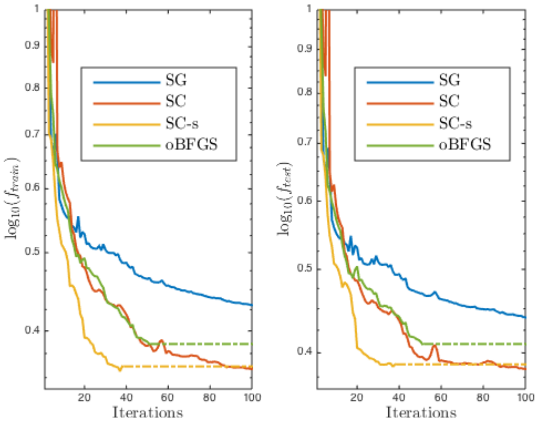

Figure 1. Training and testing losses over k for a1a for SG, SC, SC-s, and oBFGS with diminishing stepsizes.

for SC, $(\omega_2, \eta, \theta, \rho, \tau) = (1, \frac{1}{16}, 4, \frac{1}{64}, 16)$ for SC-s, and $(\omega_2, \omega_3) = (1, \frac{1}{4})$ for oBFGS. Table 2 and Figure 2 illustrate the losses achieved. One again finds that SC and SC-s terminated with low losses. The methods demonstrate somewhat unstable behavior early on due to the fixed stepsize, but ultimately still yield better results.

Table 2. Training loss, testing loss, and CPU time (in seconds) for a1a for SG, SC, SC-s, and oBFGS with a fixed stepsize.

Method	Training	Testing	CPU time (s)
SG	0.3744	0.3923	0.3355
SC	0.3383	0.3752	0.6424
SC-s	0.3650	0.3902	0.3132
oBFGS	0.3883	0.4028	0.3560

5.2. Logistic Loss, Limited-Memory Matrices

As a second test, consider again (19) with a logistic loss, but now with the rcv1(.binary) data from LIBSVM. This convex problem has $d = 47236$ and $n = 20242$. Given the larger dimension d, it is of interest to consider the performances of SG, SC-L, SC-L-s, and oLBFGS.

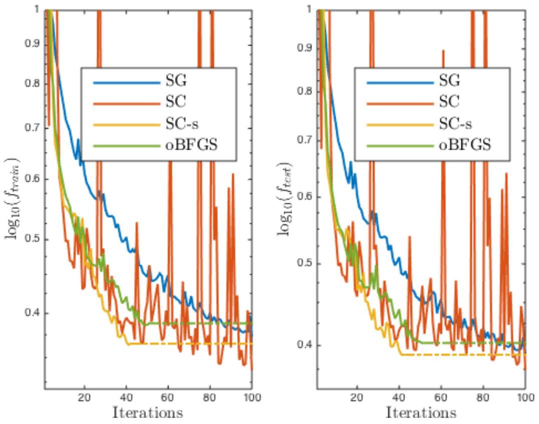

Figure 2. Training and testing losses over k for a1a for SG, SC, SC-s, and oBFGS with a fixed stepsize.

Considering stepsizes defined by (20), the lowest testing losses were found with inputs $(\omega_0, \omega_1) = (16, 1)$ for SG, $(\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (16, 1, \frac{1}{64}, 4, \frac{1}{128}, 8)$ for SC, $(\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (16, 1, \frac{1}{64}, 4, \frac{1}{128}, 16)$ for SC-s, and $(\omega_0, \omega_1, \omega_3) = (16, 1, \frac{1}{64})$ for oBFGS. Table 3 and Figure 3 illustrate losses achieved. Again, better performance is demonstrated by SC-L and SC-L-s.

Table 3. Training loss, testing loss, and CPU time for rcv1 for SG, SC-L, SC-L-s, and oLBFGS with diminishing stepsizes.

Method	Training	Testing	CPU time (s)
SG	0.7616	0.7642	25.7893
SC	0.2271	0.2403	34.4539
SC-s	0.3004	0.3068	22.5097
oBFGS	0.3108	0.3194	21.7649

Considering stepsizes defined by (21), the lowest testing losses were achieved with inputs $\omega_2 = 16$ for SG, $(\omega_2, \eta, \theta, \rho, \tau) = (16, \frac{1}{4}, 4, \frac{1}{8}, 8)$ for SC, $(\omega_2, \eta, \theta, \rho, \tau) = (16, \frac{1}{16}, 4, \frac{1}{32}, 16)$ for SC-s, and $(\omega_2, \omega_3) = (4, \frac{1}{64})$ for oBFGS. Table 4 and Figure 4 illustrate the training and testing losses achieved. Better performance is achieved by SC-L and SC-L-s. In particular, even with a fixed stepsize, SC-L performs stably enough that it performs best.

Table 4. Training loss, testing loss, and CPU time for rcv1 for SG, SC-L, SC-L-s, and oLBFGS with a fixed stepsize.

Method	Training	Testing	CPU time (s)
SG	0.2005	0.2227	25.2829
SC	0.0902	0.1339	34.3699
SC-s	0.1077	0.1478	23.8892
oBFGS	0.1374	0.1636	23.2760

5.3. Neural Network, Limited-Memory Matrices

As a final test, consider a neural network with an ℓ_2-norm loss and regularization term $\frac{1}{n}\|w\|_2^2$ with data given by the first 20000 elements from the 10-class mnist dataset. The network is defined by $x^{(j)} = s(W_j x^{(j-1)} + b_j)$ for $j \in$

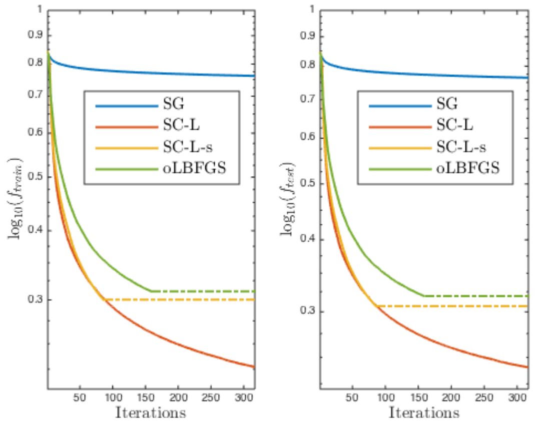

Figure 3. Training and testing losses over k for `rcv1` for `SG`, `SC-L`, `SC-L-s`, and `oLBFGS` with diminishing stepsizes.

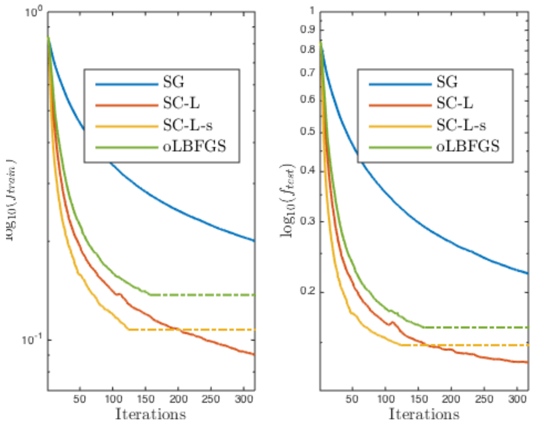

Figure 4. Training and testing losses over k for `rcv1` for `SG`, `SC-L`, `SC-L-s`, and `oLBFGS` with a fixed stepsize.

$\{1, 2, 3\}$ with activation function s defined as a componentwise sigmoid and trainable parameters (composing the decision vector w) contained in $(W_1, W_2, W_3) \in \mathbb{R}^{30\times780} \times \mathbb{R}^{100\times30} \times \mathbb{R}^{10\times100}$ and $(b_1, b_2, b_3) \in \mathbb{R}^{30} \times \mathbb{R}^{100} \times \mathbb{R}^{10}$. This nonconvex problem has $d = 27540$ and $n = 20000$. The performances of `SG`, `SC-L`, `SC-L-s`, and `oLBFGS` are of interest. While `oLBFGS` was not designed to solve nonconvex problems, it is applicable as long as $s_k^T v_k > 0$ for all $k \in \mathbb{N}$; hence, `oLBFGS` was run in this experiment, but was terminated if/when $s_k^T v_k \leq 0$ for some $k \in \mathbb{N}$.

Considering stepsizes defined by (20), the lowest testing losses were found with inputs $(\omega_0, \omega_1) = (1, 16)$ for `SG`, $(\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (1, 16, \frac{1}{16}, 4, \frac{1}{32}, 8)$ for `SC`, $(\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (1, 16, \frac{1}{16}, 4, \frac{1}{32}, 16)$ for `SC-s`, and $(\omega_0, \omega_1, \omega_3) = (16, 16, \frac{1}{64})$ for `oBFGS`. Table 5 and Figure 5 illustrate the training and testing losses achieved. As for the convex problems, `SC-L` and `SC-L-s` outperform the other methods in terms of achieving low losses. The differing initial behavior of `oLBFGS` can be explained by the fact that it terminated early (due to $s_k^T v_k \leq 0$) for the stepsize parameters that were best for the other methods; it was only able to produce low losses with much larger stepsizes that happened to slow overall performance.

Table 5. Training loss, testing loss, and CPU time for `mnist` for `SG`, `SC-L`, `SC-L-s`, and `oLBFGS` with diminishing stepsizes.

METHOD	TRAINING	TESTING	CPU TIME (S)
SG	1.6673	1.6682	164.5094
SC	1.3796	1.3862	179.4491
SC-s	1.4032	1.4072	150.7164
oBFGS	1.6550	1.6484	132.9992

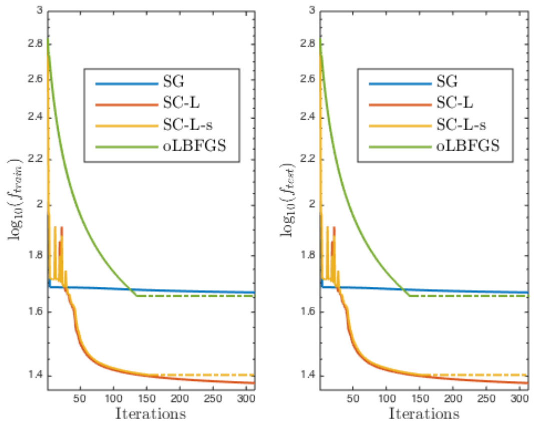

Figure 5. Training and testing losses over k for `mnist` for `SG`, `SC-L`, `SC-L-s`, and `oLBFGS` with diminishing stepsizes.

6. Conclusion

A quasi-Newton algorithm for stochastic optimization has been presented. The algorithm capitalizes on fundamental *self-correcting properties* of BFGS-type updating which, before this work, have not been exploited in the design of such an approach. The method is supported by a motivating convergence theory, which has also led to a proposed dynamic noise reduction strategy that demonstrates some practical benefits. The method with full (dense) Hessian approximations performs well for small dimensional problems, whereas, on larger-scale problems, a limited memory variant outperforms SG and another stochastic quasi-Newton method (namely, oLBFGS) while having comparable per-iteration costs. Numerical experiments on machine learning test problems—two convex and one nonconvex—have demonstrated that the proposed variants of the method are effective in terms of achieving low training and testing losses, even within a first epoch.

Acknowledgements

The author is grateful to the anonymous referees for their valuable comments and suggestions about the paper. He is also grateful to Jorge Nocedal and Daniel P. Robinson for insightful conversations that led to the inception and refinement of the work that led to this paper.

References

Bertsekas, D. P. *Convex Optimization Algorithms*. Athena Scientific, Nashua, NH, USA, 2015.

Bordes, A., Bottou, L., and Gallinari, P. SGD-QN: Careful Quasi-Newton Stochastic Gradient Descent. *Journal of Machine Learning Research*, 10:1737–1754, 2009.

Bottou, L. Online algorithms and stochastic approximations. In Saad, D. (ed.), *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK, 1998. URL http://leon.bottou.org/papers/bottou-98x. revised, Oct. 2012.

Bottou, L. Large-Scale Machine Learning with Stochastic Gradient Descent. In Lechevallier, Y. and Saporta, G. (eds.), *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT 2010)*, pp. 177–187, Paris, France, 2010. Springer.

Bottou, L. and Bousquet, O. The Tradeoffs of Large Scale Learning. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T. (eds.), *Advances in Neural Information Processing Systems 20*, pp. 161–168. Curran Associates, Inc., 2008.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization Methods for Large-Scale Machine Learning. Working Paper, to be submitted to SIAM Review, 2016.

Broyden, C. G. The Convergence of a Class of Double-Rank Minimization Algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6(1):76–90, 1970.

Byrd, R. H. and Nocedal, J. A Tool for the Analysis of Quasi-Newton Methods with Application to Unconstrained Minimization. *SIAM Journal on Numerical Analysis*, 26(3):727–739, 1989.

Byrd, R. H., Nocedal, J., and Yuan, Y. Global Convergence of a Class of Quasi-Newton Methods on Convex Problems. *SIAM Journal on Numerical Analysis*, 24(5): 1171–1189, 1987.

Byrd, R. H., Hansen, S. L., Nocedal, J., and Singer, Y. A Stochastic Quasi-Newton Method for Large-Scale Optimization. *arXiv.org*, 2015. URL http://arxiv.org/abs/1401.7020.

Davidon, W. C. Variable Metric Method for Minimization. *SIAM Journal on Optimization*, 1(1):1–17, 1991.

Defazio, A., Bach, F. R., and Lacoste-Julien, S. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. *Computing Research Repository*, abs/1407.0202, 2014. URL http://arxiv.org/abs/1407.0202.

Dennis, J. E. and Moré, J. J. A Characterization of Superlinear Convergence and Its Application to Quasi-Newton Methods. *Mathematics of Computation*, 28(126):549–560, 1974.

Fletcher, R. A New Approach to Variable Metric Algorithms. *Computer Journal*, 13(3):317–322, 1970.

Fletcher, R. and Powell, M. J. D. A Rapidly Convergent Descent Method for Minimization. *Computer Journal*, 6 (2):163–168, 1963.

Goldfarb, D. A Family of Variable Metric Updates Derived by Variational Means. *Mathematics of Computation*, 24 (109):23–26, 1970.

Johnson, R. and Zhang, T. Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 315–323. Curran Associates, Inc., 2013. URL http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips26/238.pdf.

Mokhtari, A. and Ribeiro, A. RES: Regularized Stochastic BFGS Algorithm. *IEEE Transactions on Signal Processing*, 62(23):6089–6104, 2014.

Nocedal, J. Updating Quasi-Newton Matrices With Limited Storage. *Mathematics of Computation*, 35(151): 773–782, 1980.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, Second edition, 2006.

Pearson, J. D. Variable Metric Methods of Minimisation. *The Computer Journal*, 12(2):171–178, 1969.

Powell, M. J. D. Some Global Convergence Properties of a Variable Metric Algorithm for Minimization with Exact Line Searches. In Cottle, R. W. and Lemke, C. E. (eds.), *Nonlinear Programming, SIAM-AMS Proceedings, Vol. IX*. American Mathematical Society, 1976.

Powell, M. J. D. Algorithms for Nonlinear Constraints that use Lagrangian Functions. *Mathematical Programming*, 14(1):224–248, 1978.

Ritter, K. Local and Superlinear Convergence of a Class of Variable Metric Methods. *Computing*, 23(3):287–297, 1979.

Ritter, K. Global and Superlinear Convergence of a Class of Variable Metric Methods. In König, H., Korte, B., and Ritter, K. (eds.), *Mathematical Programming at Oberwolfach*, pp. 178–205. Springer Berlin Heidelberg, 1981.

Robbins, H. and Monro, S. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3): 400–407, 1951.

Schraudolph, Nicol N., Yu, Jin, and Günter, Simon. A Stochastic Quasi-Newton Method for Online Convex Optimization. In Meila, Marina and Shen, Xiaotong (eds.), *Proc. 11th Intl. Conf. Artificial Intelligence and Statistics (AIstats)*, volume 2 of *Workshop and Conference Proceedings*, pp. 436–443, San Juan, Puerto Rico, 2007. Journal of Machine Learning Research.

Shanno, D. F. Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computation*, 24(111):647–656, 1970.

Wang, X., Ma, S., and Liu, W. Stochastic Quasi-Newton Methods for Nonconvex Stochastic Optimization. *arXiv.org*, 2015. URL http://arxiv.org/abs/1412.1196.

Werner, J. Über die globale Konvergenz von Variable-Metrik-Verfahren mit nicht-exakter Schrittweitenbestimmung. *Numerische Mathematik*, 31(3):321–334, 1978.

A. Proofs

Proof of Theorem 1. (This proof follows that of Thm. 2.1 in (Byrd & Nocedal, 1989).) Defining, for all $k \in \mathbb{N}$,

$$\begin{aligned}\zeta_k := &-\ln(\cos^2\delta_k) \\ &-\left(1 - \frac{\epsilon_k}{\cos^2\delta_k} + \ln\left(\frac{\epsilon_k}{\cos^2\delta_k}\right)\right) \geq 0, \end{aligned} \qquad (22)$$

one has by (8) and (9) that

$$\gamma(H_{k+1}) \leq \gamma(H_k) + \theta - 1 - \ln\eta - \zeta_k,$$

from which it follows that, for any $K \in \mathbb{N}$,

$$\gamma(H_{K+1}) \leq \gamma(H_1) + (\theta - 1 - \ln\eta)K - \sum_{k=1}^{K}\zeta_k.$$

Since $\gamma(H_{K+1}) > 0$ and $\gamma(H_1) > 0$, this implies that

$$\frac{1}{K}\sum_{k=1}^{K}\zeta_k < \gamma(H_1) + (\theta - 1 - \ln\eta). \qquad (23)$$

Now consider a fixed $K \in \mathbb{N}$, let $J_{p,K}$ be the set of indices corresponding to the $\lceil pK \rceil$ smallest elements of ζ_k over $k \in \{1,\ldots,K\}$, and let $\zeta_{p,K}$ denote the largest ζ_k over $k \in J_{p,K}$. Then, it follows that

$$\begin{aligned}\frac{1}{K}\sum_{k=1}^{K}\zeta_k &\geq \frac{1}{K}\left(\zeta_{p,K} + \sum_{k=1, k\notin J_{p,K}}^{K}\zeta_k\right) \\ &\geq \left(\frac{1 + K - \lceil pK \rceil}{K}\right)\zeta_{p,K} \\ &\geq (1-p)\zeta_{p,K},\end{aligned}$$

which along with (23) implies that, for all $k \in J_{p,K}$,

$$\zeta_k \leq \frac{1}{1-p}(\gamma(H_1) + \theta - 1 - \ln\eta) =: \psi_0. \qquad (24)$$

Since (22) implies that $\zeta_k \geq -\ln(\cos^2\delta_k)$ for all $k \in J_{p,K}$, one has from (24) that $-\ln(\cos^2\delta_k) < \psi_0$ for all $k \in J_{p,K}$, which means that $\cos\delta_k > e^{-\psi_0/2} =: \psi_1$ for all $k \in J_{p,K}$, meaning that the first inequality in (10) holds with $\iota := \psi_1$. Likewise, (22) and (24) imply that

$$1 - \frac{\epsilon_k}{\cos^2\delta_k} + \ln\left(\frac{\epsilon_k}{\cos^2\delta_k}\right) > -\psi_0.$$

for all $k \in J_{p,K}$. Hence, since

$$1 - \zeta + \ln(\zeta) \searrow -\infty \ \text{ as } \ \zeta \searrow 0 \ \text{ or } \ \zeta \nearrow \infty.$$

it follows that there exist positive constants $\psi_2 \in \mathbb{R}_{++}$ and $\psi_3 \in \mathbb{R}_{++}$ such that

$$\psi_2 \leq \frac{\epsilon_k}{\cos^2\delta_k} \leq \psi_3,$$

for all $k \in J_{p,K}$, which means, with $\cos^2\delta_k > \psi_1$, that

$$\psi_4 := \psi_1^2\psi_2 \leq \epsilon_k \leq \psi_3$$

for all $k \in J_{p,K}$. Since $\|H_k s_k\|_2/\|s_k\|_2 = \epsilon_k/\cos\delta_k$, it follows that, for all $k \in J_{p,K}$, one has

$$\psi_4 \leq \frac{\|H_k s_k\|_2}{\|s_k\|_2} \leq \frac{\psi_3}{\psi_1},$$

meaning that the latter inequalities in (10) hold with the constants $\kappa := \psi_4$ and $\lambda := \psi_3/\psi_1$. □

Proof of Corollary 2. By (2) and since $M_k = H_k^{-1}$ for all $k \in \mathbb{N}$, the inequalities in (10) imply

$$\iota \leq \frac{g_k^T M_k g_k}{\|M_k g_k\|_2\|g_k\|_2} \ \text{ and } \ \kappa \leq \frac{\|g_k\|_2}{\|M_k g_k\|_2} \leq \lambda.$$

From the first and third of these inequalities, it follows that

$$g_k^T M_k g_k \geq \iota\|M_k g_k\|_2\|g_k\|_2 \geq \frac{\iota}{\lambda}\|g_k\|_2^2,$$

while, from the second of these inequalities, it follows that

$$\|M_k g_k\|_2^2 \le \frac{1}{\kappa^2}\|g_k\|_2^2.$$

Hence, (11) holds for $\mu := \iota/\lambda$ and $\nu := \kappa^{-2}$. □

The remaining results and proofs are based on the analysis in the working paper (Bottou et al., 2016).

Proof of Lemma 4. For any arbitrary $k \in \mathbb{N}$, it follows by (12) and Prop. 6.1.2 in (Bertsekas, 2015) that

$$\begin{aligned} & f(w_{k+1}) - f(w_k) \\ & \le \nabla f(w_k)^T(w_{k+1} - w_k) + \tfrac{1}{2}L\|w_{k+1} - w_k\|_2^2 \\ & = -\alpha_k \nabla f(w_k)^T M_k g_k + \tfrac{1}{2}\alpha_k^2 L\|M_k g_k\|_2^2. \end{aligned}$$

Taking expectations with respect to the distribution of ξ_k and observing (13) and (14), it follows that

$$\begin{aligned} & \mathbb{E}_{\xi_k}[f(w_{k+1})] - f(w_k) \\ \le & -\alpha_k \nabla f(w_k)^T \mathbb{E}_{\xi_k}[M_k g_k] + \tfrac{1}{2}\alpha_k^2 L \mathbb{E}_{\xi_k}[\|M_k g_k\|_2^2] \\ \le & -\alpha_k \rho \|\nabla f(w_k)\|_2^2 + \tfrac{1}{2}\alpha_k^2 L(\sigma + \tau\|\nabla f(w_k)\|_2^2) \\ \le & -\alpha_k(\rho - \tfrac{1}{2}\alpha_k L\tau)\|\nabla f(w_k)\|_2^2 + \tfrac{1}{2}\alpha_k^2 L\sigma, \end{aligned}$$

as desired. □

The proof of Theorem 5 requires the following two technical lemmas, which are easily proved.

Lemma 7. *Let $\{\alpha_k\}$ satisfy (15). Then, for $\{L, \rho, \tau\}$ defined in Assumption 3 and all sufficiently large $k \in \mathbb{N}$,*

$$\alpha_k(\rho - \tfrac{1}{2}\alpha_k L\tau) \ge \tfrac{1}{2}\alpha_k\rho.$$

Proof. To derive a contradiction, suppose that there is an infinite subsequence of stepsizes indexed by $\mathcal{K} \subseteq \mathbb{N}$ with

$$\alpha_k(\rho - \tfrac{1}{2}\alpha_k L\tau) < \tfrac{1}{2}\alpha_k\rho \iff \tfrac{1}{2}\alpha_k\rho < \tfrac{1}{2}\alpha_k^2 L\tau. \quad (25)$$

Summing over $k \in \mathcal{K}$, one obtains

$$\tfrac{1}{2}\rho \sum_{k\in\mathcal{K}} \alpha_k < \tfrac{1}{2}L\tau \sum_{k\in\mathcal{K}} \alpha_k^2.$$

If the left-hand side of this inequality were infinite, then the right-hand side must be infinite. However, that would contradict the second condition in (15). Hence, the sum on the left-hand side must be finite, from which it follows that $\{\alpha_k\}_{k\in\mathcal{K}} \to 0$. But this contradicts the definition of $\mathcal{K}$ since (25) has $\alpha_k > \rho/(L\tau)$ for all $k \in \mathcal{K}$. □

For the next lemma, $[\upsilon]_+ := \max\{\upsilon, 0\}$ for any $\upsilon \in \mathbb{R}$.

Lemma 8. *Let $\{\phi_k\} \subset \mathbb{R}_+$ and $\{\chi_k\} \subset \mathbb{R}_+$ be nonnegative scalar sequences such that, for all $k \in \mathbb{N}$,*

$$\sum_{k=0}^{\infty} \phi_k < \infty \ \text{ and } \ [\chi_{k+1} - \chi_k]_+ \le \phi_k. \quad (26)$$

Then, the sequence $\{\chi_k\}$ is convergent.

Proof. Let $\Delta\chi_k^+ := \sum_{j=1}^k [\chi_{j+1} - \chi_j]_+$ and $\Delta\chi_k^- := \sum_{j=1}^k [\chi_j - \chi_{j+1}]_+$ for all $k \in \mathbb{N}$, which by definition implies that $\{\Delta\chi_k^+\}$ and $\{\Delta\chi_k^-\}$ are nonnegative, and, by (26), the former is convergent. Hence, since

$$\chi_1 + \Delta\chi_k^+ - \Delta\chi_k^- = \chi_{k+1} \ge 0 \quad (27)$$

for all $k \in \mathbb{N}$, the sequence $\{\Delta\chi_k^-\}$ is bounded above by the convergent sequence $\{\chi_1 + \Delta\chi_k^+\}$, meaning that it is convergent. Since both $\{\Delta\chi_k^+\}$ and $\{\Delta\chi_k^-\}$ are convergent, it follows by (27) that $\{\chi_k\}$ is convergent. □

Proof of Theorem 5. By Lemma 7, the first term on the right-hand side of the concluding inequality in Lemma 4 is nonpositive for all large $k \in \mathbb{N}$. Hence, without loss of generality, one may assume that for all $k \in \mathbb{N}$ one has

$$(\mathbb{E}_{\xi_k}[f(w_{k+1})] - f_{\min}) - (f(w_k) - f_{\min}) \le \tfrac{1}{2}\alpha_k^2 L\sigma. \quad (28)$$

Now taking total expectations, it follows that

$$\begin{aligned} & \mathbb{E}[f(w_{k+1}) - f_{\min}] - \mathbb{E}[f(w_k) - f_{\min}] \\ & = \mathbb{E}[(\mathbb{E}_{\xi_k}[f(w_{k+1})] - f_{\min}) - (f(w_k) - f_{\min})] \\ & \le \tfrac{1}{2}\alpha_k^2 L\sigma. \end{aligned}$$

It now follows from Lemma 8 with $\{\phi_k\} := \{\tfrac{1}{2}\alpha_k^2 L\sigma\}$ and $\{\chi_k\} := \{\mathbb{E}[f(w_k) - f_{\min}]\}$—where the inequalities in (26) follow from (15) and the last inequality above—that $\{\mathbb{E}[f(w_k) - f_{\min}]\}$, and hence $\{\mathbb{E}[f(w_k)]\}$, is convergent.

Now return to the result of Lemma 4, where taking total expectations and summing over the first K iterations yields

$$\begin{aligned} & \mathbb{E}[f(w_{K+1}) - f_{\min}] - \mathbb{E}[f(w_1) - f_{\min}] \\ & \le -\sum_{k=1}^{K} \alpha_k(\rho - \tfrac{1}{2}\alpha_k L\tau)\|\nabla f(w_k)\|_2^2 + \tfrac{1}{2}L\sigma\sum_{k=1}^{K}\alpha_k^2. \end{aligned}$$

Since the left-hand side of this expression is convergent as $K \to \infty$, it follows that the right-hand side is bounded below as $K \to \infty$. Then, since (15) ensures that the last sum on the right-hand side is convergent as $K \to \infty$, it follows along with the result of Lemma 7 that

$$\sum_{k=1}^{K} \alpha_k \|\nabla f(w_k)\|_2^2$$

is convergent as $K \to \infty$, which is (16). It then follows from (16) that (17) holds. □

Proof of Theorem 6. First, suppose that $\{\alpha_k\}$ is defined as in part (a) of the theorem. As is well known, since f is c-strongly convex, it follows that, for all $w \in \mathbb{R}^d$,

$$2c(f(w) - f(w_*)) \leq \|\nabla f(w)\|_2^2. \tag{29}$$

Using this fact, the restriction on the stepsize, and Lemma 4, it follows that for all $k \in \mathbb{N}$ one has

$$\begin{aligned}
&\mathbb{E}_{\xi_k}[f(w_{k+1}) - f(w_k)] \\
\leq\; & -\alpha(\rho - \tfrac{1}{2}\alpha L\tau)\|\nabla f(w_k)\|_2^2 + \tfrac{1}{2}\alpha^2 L\sigma \\
\leq\; & -\tfrac{1}{2}\alpha\rho\|\nabla f(w_k)\|_2^2 + \tfrac{1}{2}\alpha^2 L\sigma \\
\leq\; & -\alpha c\rho(f(w_k) - f(w_*)) + \tfrac{1}{2}\alpha^2 L\sigma.
\end{aligned}$$

Taking total expectations, this yields

$$\begin{aligned}
&\mathbb{E}[f(w_{k+1}) - f(w_*)] \\
&\leq (1 - \alpha c\rho)\mathbb{E}[f(w_k) - f(w_*)] + \tfrac{1}{2}\alpha^2 L\sigma.
\end{aligned}$$

Subtracting $\frac{\alpha L\sigma}{2c\rho}$ from both sides and rearranging,

$$\begin{aligned}
&\mathbb{E}[f(w_{k+1}) - f(w_*)] - \frac{\alpha L\sigma}{2c\rho} \\
&\leq (1 - \alpha c\rho)\mathbb{E}[f(w_k) - f(w_*) + \frac{\alpha^2 L\sigma}{2} - \frac{\alpha L\sigma}{2c\rho} \\
&\leq (1 - \alpha c\rho)\left(\mathbb{E}[f(w_k) - f(w_*)] - \frac{\alpha L\sigma}{2c\rho}\right),
\end{aligned}$$

which leads to the desired conclusion.

Now suppose that $\{\alpha_k\}$ is defined as in part (b). By the definition of $k(\alpha)$, the restriction on the initial stepsize, Lemma 4, and (29), it follows that, for all $k \in \mathbb{N}$,

$$\begin{aligned}
&\mathbb{E}_{\xi_k}[f(w_{k+1}) - f(w_k)] \\
\leq\; & -\alpha_k(\rho - \tfrac{1}{2}\alpha_k L\tau)\|\nabla f(w_k)\|_2^2 + \tfrac{1}{2}\alpha_k^2 L\sigma \\
\leq\; & -\tfrac{1}{2}\alpha_k\rho\|\nabla f(w_k)\|_2^2 + \tfrac{1}{2}\alpha_k^2 L\sigma \\
\leq\; & -\alpha_k c\rho(f(w_k) - f(w_*)) + \tfrac{1}{2}\alpha_k^2 L\sigma.
\end{aligned}$$

Taking total expectations, this yields

$$\begin{aligned}
&\mathbb{E}[f(w_{k+1}) - f(w_*)] \\
&\leq (1 - \alpha_k c\rho)\mathbb{E}[f(w_k) - f(w_*)] + \tfrac{1}{2}\alpha_k^2 L\sigma.
\end{aligned}$$

The desired conclusion can now be reached through an inductive argument. By the definition of $C(\alpha)$, the inequality (18) holds for $k = k(\alpha)$. Assuming that it holds for some $k \geq k(\alpha)$, it follows from the conclusion above that

$$\begin{aligned}
&\mathbb{E}[f(w_{k+1}) - f(w_*)] \\
&\leq \left(1 - \frac{\alpha c\mu}{k}\right)\frac{C(\alpha)}{k} + \frac{\alpha^2 L\sigma}{2k^2} \\
&\leq \left(\frac{k - \alpha c\mu}{k^2}\right)C(\alpha) + \frac{\alpha^2 L\sigma}{2k^2} \\
&\leq \left(\frac{k-1}{k^2}\right)C(\alpha) - \left(\frac{\alpha c\rho - 1}{k^2}\right)C(\alpha) + \frac{\alpha^2 L\sigma}{2k^2} \\
&\leq \frac{C(\alpha)}{k+1},
\end{aligned}$$

where the last inequality follows by the definition of $C(\alpha)$ and the fact that $k^2 \geq (k+1)(k-1)$. □