A Self-Correcting Variable-Metric Algorithm for Stochastic Optimization

Frank E. Curtis
Department of ISE, Lehigh University, 200 W. Packer Ave., Bethlehem, PA 18015 USA

FRANK.E.CURTIS@GMAIL.COM

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

A Self-Correcting Variable-Metric Algorithm for Stochastic Optimization

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 \textit{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 \textit{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 \textit{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 \textit{skipping} and \textit{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.

- 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 function \( f : \mathbb{R}^d \to \mathbb{R} \) defined by the expectation, in terms of the distribution of a random variable \( \xi \) with domain \( \Xi \), of a stochastic function \( F : \mathbb{R}^d \times \Xi \to \mathbb{R} \):

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

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

\[
s_k \leftarrow -\alpha_k M_k g_k
\]

then

\[
w_{k+1} \leftarrow w_k + s_k,
\]

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
A Self-Correcting Variable-Metric Algorithm for Stochastic Optimization

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 \( H > 0 \), consider \( \gamma : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R} \) 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 \( k \in \mathbb{N} \),

\[
\text{trace}(H_{k+1}) = \text{trace}(H_k) - \frac{\|H_k s_k\|_2^2}{s_k^T H_k s_k} + \frac{\|v_k\|_2^2}{s_k^T v_k}.
\]

and

\[
\det(H_{k+1}) = \det(H_k) \left( \frac{s_k^T v_k}{s_k^T H_k s_k} \right),
\]

(Pearson, 1969) with which one can relate \( \gamma(H_{k+1}) \) and \( \gamma(H_k) \). Specifically, assuming that \( H_k > 0 \) 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

\[
\gamma(H_{k+1}) = \gamma(H_k) + \frac{\|v_k\|_2^2}{s_k^T v_k} - 1 - \ln \left( \frac{s_k^T v_k}{\|s_k\|_2^2} \right) + \ln(\cos^2 \delta_k) + \frac{1 - \epsilon_k}{\cos^2 \delta_k} + \ln \left( \frac{\epsilon_k}{\cos^2 \delta_k} \right). \tag{8}
\]

Nonpositivity of the latter two terms follows since \( s_k \neq 0 \) and \( H_k > 0 \) 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 \( \gamma \) 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

\[
\eta \leq \frac{s_k^T v_k}{\|s_k\|_2^2} \quad \text{and} \quad \frac{\|v_k\|_2^2}{s_k^T v_k} \leq \theta. \tag{9}
\]

Then, for any \( p \in (0, 1) \), there exist constants \( \{\epsilon, \kappa, \lambda\} \subset \mathbb{R}^d \) such that
such that, for any \( K \in \{2, 3, \ldots\} \), the following relations hold for at least \( \lfloor pK \rfloor \) values of \( k \in \{1, \ldots, K\} \):
\[
\ell \leq \frac{s_k^T H_k s_k}{\| s_k \|_2 \| H_k s_k \|_2} \quad \text{and} \quad \kappa \leq \frac{\| H_k s_k \|_2}{\| s_k \|_2} \leq \lambda. \quad (10)
\]
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 \( \{\mu, \nu\} \subseteq \mathbb{R}_{++} \) such that, for any \( K \in \{2, 3, \ldots\} \), the following relations hold for at least \( \lfloor pK \rfloor \) values of \( k \in \{1, \ldots, K\} \):
\[
\mu \|g_k\|_2^2 \leq g_k^T M_k g_k \quad \text{and} \quad \|M_k g_k\|_2^2 \leq \nu \|g_k\|_2^2. \quad (11)
\]

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 \( \xi_k \) taking values in a set \( \Xi \) that does not depend on \( \{w_j\}_{j=1}^{k-1} \). In what follows, \( \mathbb{E}_\xi[\cdot] \) denotes expectation taken with respect to the distribution of \( \xi_k \) 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).)

**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 \( W \subseteq \mathbb{R}^d \) 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 \( \{w, \overline{w}\} \subseteq W \)
\[
\|\nabla f(w) - \nabla f(\overline{w})\|_2 \leq L \|w - \overline{w}\|_2. \quad (12)
\]
In addition, there exists a scalar \( \rho \in \mathbb{R}_{++} \) such that
\[
\rho \|\nabla f(w_k)\|_2^2 \leq \nabla f(w_k)^T \mathbb{E}_\xi[M_k g_k] \quad (13)
\]
and there exist scalars \( \sigma \in \mathbb{R}_{++} \) and \( \tau \in \mathbb{R}_{++} \) such that
\[
\mathbb{E}_\xi[\|M_k g_k\|_2^2] \leq \sigma + \tau \|\nabla f(w_k)\|_2^2 \quad (14)
\]
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 \( k \in \mathbb{N} \):
\[
\mathbb{E}_\xi[f(w_{k+1})] - f(w_k) \leq -\alpha_k \nabla f(w_k)^T \mathbb{E}_\xi[M_k g_k] + \frac{1}{2} \alpha_k^2 \mathbb{E}_\xi[\|M_k g_k\|_2^2]
\]
\[
\leq -\alpha_k (\rho - \frac{1}{2} \alpha_k L \tau) \|\nabla f(w_k)\|_2^2 + \frac{1}{2} \alpha_k^2 L \sigma.
\]
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
A Self-Correcting Variable-Metric Algorithm for Stochastic Optimization

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

\[
\sum_{k=1}^{\infty} \alpha_k = \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty. \tag{15}
\]

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

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

which, in particular, implies that

\[\liminf_{k \to \infty} E[\|\nabla f(w_k)\|^2] = 0. \tag{17}\]

The next result addresses the minimization of a strongly convex objective, which is guaranteed to have a unique minimizer, call it \( w_\ast \in \mathbb{R}^d \). Two example stepsize sequences are considered, though results similar to the latter can also be proved for related types of \( O(\frac{1}{k}) \) 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 \( \alpha \in (0, \min\{\frac{1}{2c\rho}, \frac{1}{c\rho}\}) \) for all \( k \in \mathbb{N} \), then the expected optimality gap satisfies the following for all \( k \in \mathbb{N} \):

\[
E[f(w_k)] - f(w_\ast) \leq \frac{\alpha L\sigma}{2c\rho} + (1 - \alpha c\rho)^k \left( f(w_1) - f(w_\ast) - \frac{\alpha L\sigma}{2c\rho} \right) \rightarrow \frac{\alpha L\sigma}{2c\rho} \quad \text{(as } k \to \infty) \tag{18}.
\]

(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 \( k \geq k(\alpha) := \frac{c\rho}{\alpha} + 1 \):

\[
E[f(w_k)] - f(w_\ast) \leq \frac{C(\alpha)}{k}, \tag{18}
\]

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

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

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 \( E[\xi_k g_k] = \nabla f(w_k) \) 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 \( g_k = \nabla f(w_k) \) 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)

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

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 \( \hat{k}_{\text{max}} \)) and one
A Self-Correcting Variable-Metric Algorithm for Stochastic Optimization

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 \( \{ y_j \}_{j=k-m+1}^k \) 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 \( O(d) \) (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, 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

\[
f(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w; x_i, y_i),
\]

where \( \{ (x_i, y_i) \}_{i=1}^{n} \) represents training data and \( \ell \) 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

\[
\alpha_k = \omega_0/(\omega_1 + k) \quad \text{for all } k \in \mathbb{N}
\]

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.,

\[
\alpha_k = \omega_2 \quad \text{for all } k \in \mathbb{N},
\]

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 algorithms, 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 \( \omega_{\text{max}} = 2 \) and \( \sigma = 0 \) were fixed. For oBFGS and oLBFGS, following (Schraudolph et al., 2007), the stochastic gradient displacement vectors were computed for the sample \( S_k \) in iteration \( k \in \mathbb{N} \) as

\[
\frac{1}{|S_k|} \sum_{i \in S_k} \left( \nabla \ell(w_{k+1}; x_i, y_i) - \nabla \ell(w_k; x_i, y_i) \right) + \omega_3 s_k.
\]

The values \( \omega_3 \in \{2^{-6}, 2^{-4}, 2^{-2}, 2^{-4}\} \) 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 8 GB 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 a 1a 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, \( (\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (16, 16, \frac{1}{2}, 4, \frac{1}{3}, 8) \) 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, \( (\omega_2, \eta, \theta, \rho, \tau) = (4, \frac{1}{2}, 1, 4, \frac{1}{2}\tau, 8) \)

\(^1\)http://coral.ie.lehigh.edu/frankecurtis/software/
\(^2\)https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
for a1a

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

<table>
<thead>
<tr>
<th>METHOD</th>
<th>TRAINING</th>
<th>TESTING</th>
<th>CPU TIME (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG</td>
<td>0.4305</td>
<td>0.4398</td>
<td>0.3405</td>
</tr>
<tr>
<td>SC</td>
<td>0.3588</td>
<td>0.3832</td>
<td>0.5389</td>
</tr>
<tr>
<td>SC-s</td>
<td>0.3614</td>
<td>0.3879</td>
<td>0.3705</td>
</tr>
<tr>
<td>oBFGS</td>
<td>0.3853</td>
<td>0.4096</td>
<td>0.4191</td>
</tr>
</tbody>
</table>

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

for SC, \((\omega_2, \eta, \theta, \rho, \tau) = (1, 1, 4, 1, 16)\) for SC-s, and \((\omega_2, \omega_3) = (1, \frac{1}{2})\) 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.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>TRAINING</th>
<th>TESTING</th>
<th>CPU TIME (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG</td>
<td>0.3744</td>
<td>0.3923</td>
<td>0.3355</td>
</tr>
<tr>
<td>SC</td>
<td>0.3383</td>
<td>0.3752</td>
<td>0.6424</td>
</tr>
<tr>
<td>SC-s</td>
<td>0.3650</td>
<td>0.3902</td>
<td>0.3132</td>
</tr>
<tr>
<td>oBFGS</td>
<td>0.3883</td>
<td>0.4028</td>
<td>0.3560</td>
</tr>
</tbody>
</table>

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.

Considering stepizes defined by (20), the lowest testing losses were found with inputs \((\omega_0, \omega_1) = (16, 1)\) for SG, \((\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (16, 1, \frac{1}{16}, 4, 1, 16)\) for SC, \((\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (16, 1, \frac{1}{16}, 4, 1, 16)\) for SC-s, and \((\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (16, 1, \frac{1}{16}, 4, 1, 16)\) for SC-L, and \((\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (16, 1, \frac{1}{16}, 8)\) for SC-s, and \((\omega_0, \omega_1, \eta, \theta, \rho, \tau) = (16, 1, \frac{1}{16}, 4, 1, 16)\) for oBFGS. Table 3 and Figure 3 illustrate losses achieved. Again, better performance is demonstrated by SC-L and SC-L-s. In particular, even with a fixed stepsize, SC-L performs stably enough that it performs best.

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

<table>
<thead>
<tr>
<th>METHOD</th>
<th>TRAINING</th>
<th>TESTING</th>
<th>CPU TIME (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG</td>
<td>0.7616</td>
<td>0.7642</td>
<td>25.7893</td>
</tr>
<tr>
<td>SC</td>
<td>0.2271</td>
<td>0.2403</td>
<td>34.4539</td>
</tr>
<tr>
<td>SC-s</td>
<td>0.3004</td>
<td>0.3068</td>
<td>22.5097</td>
</tr>
<tr>
<td>oBFGS</td>
<td>0.3108</td>
<td>0.3194</td>
<td>21.7649</td>
</tr>
</tbody>
</table>

Considering stepizes defined by (21), the lowest testing losses were found with inputs \(\omega_2 = 16\) for SG, \((\omega_2, \eta, \theta, \rho, \tau) = (16, 1, \frac{1}{4}, 4, 1, 8)\) for SC, \((\omega_2, \eta, \theta, \rho, \tau) = (16, 1, \frac{1}{4}, 4, 1, 16)\) for SC-s, and \((\omega_2, \omega_3) = (4, \frac{1}{16})\) 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.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>TRAINING</th>
<th>TESTING</th>
<th>CPU TIME (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG</td>
<td>0.2005</td>
<td>0.2227</td>
<td>25.2829</td>
</tr>
<tr>
<td>SC</td>
<td>0.0902</td>
<td>0.1339</td>
<td>34.4539</td>
</tr>
<tr>
<td>SC-s</td>
<td>0.2271</td>
<td>0.2403</td>
<td>34.4539</td>
</tr>
<tr>
<td>oBFGS</td>
<td>0.3108</td>
<td>0.3194</td>
<td>21.7649</td>
</tr>
</tbody>
</table>

5.3. Neural Network, Limited-Memory Matrices

As a final test, consider a neural network with an \(\ell_2\)-norm loss and regularization term \(\frac{1}{2}\|w\|^2\) 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 \(j \in \mathbb{N}^+\).
A Self-Correcting Variable-Metric Algorithm for Stochastic Optimization

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.

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


Since $\gamma$ one has by (8) and (9) that

$$\frac{1}{K} \sum_{k=1}^{K} \zeta_k \geq \frac{1}{K} \left( \zeta_{p,K} + \sum_{k=1, k \notin J_p,K}^{K} \zeta_k \right) \geq \left( 1 + K - \lfloor pK \rfloor \right) \zeta_{p,K} \geq \left( 1 - \frac{p}{K} \right) \zeta_{p,K},$$

which along with (23) implies that, for all $k \in J_p,K$,

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

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

$$\frac{1}{K} \sum_{k=1}^{K} \zeta_k \geq \frac{1}{K} \left( \zeta_{p,K} + \sum_{k=1, k \notin J_p,K}^{K} \zeta_k \right) \geq \left( 1 + K - \lfloor pK \rfloor \right) \zeta_{p,K} \geq \left( 1 - \frac{p}{K} \right) \zeta_{p,K},$$

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

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

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

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

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

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 $\psi := \psi_1$. 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

$$1 - \zeta + \ln(\zeta) \downarrow -\infty$$

as $\zeta \downarrow 0$ 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 $\frac{\|H_k s_k\|_2}{\|s_k\|_2} = \epsilon_k / \cos \delta_k$, 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 $\lambda := \psi_3/\psi_1$. \qed

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

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

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

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

and

$$\frac{\|g_k\|_2}{\|s_k\|_2} \leq \frac{\epsilon}{\lambda}.$$
while, from the second of these inequalities, it follows that
\[ \|M_k g_k\|^2 \leq \frac{1}{\kappa^2} \|g_k\|^2. \]

Hence, (11) holds for \( \mu := \nu / \lambda \) 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
\[
f(k_{n+1}) - f(k_n) \\
\leq \nabla f(k_n)^T (k_{n+1} - k_n) + \frac{1}{2} L \|k_{n+1} - k_n\|^2 \\
= - \alpha_n \nabla f(k_n)^T M_n g \xi_n + \frac{1}{2} \alpha_n^2 L \|M_n g\|^2.
\]

Taking expectations with respect to the distribution of \( \xi_n \) and observing (13) and (14), it follows that
\[
E \xi_n[f(k_{n+1})] - f(k_n) \\
\leq - \alpha_n \nabla f(k_n)^T E \xi_n[M_n g] + \frac{1}{2} \alpha_n^2 L E \xi_n\|M_n g\|^2 \\
\leq - \alpha_n \|\nabla f(k_n)\|^2 + \frac{1}{2} \alpha_n^2 L(\sigma + \tau) \|\nabla f(k_n)\|^2 \\
\leq - \alpha_n (\rho - \frac{1}{2} \alpha_n L \sigma) \|\nabla f(k_n)\|^2 + \frac{1}{2} \alpha_n^2 L \sigma,
\]
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 \( k \in \mathbb{N} \),
\[ \alpha_k (\rho - \frac{1}{2} \alpha_n L \tau) \geq \frac{1}{2} \alpha_k \rho. \]

**Proof.** To derive a contradiction, suppose that there is an infinite subsequence of stepsizes indexed by \( K \subseteq \mathbb{N} \) with
\[ \alpha_k (\rho - \frac{1}{2} \alpha_n L \tau) < \frac{1}{2} \alpha_k \rho \iff \frac{1}{2} \alpha_k \rho < \frac{1}{2} \alpha_n^2 L \tau. \] (25)

Summing over \( k \in K \), one obtains
\[ \frac{1}{2} \rho \sum_{k=1}^{K} \alpha_k < \frac{1}{2} L \tau \sum_{k=1}^{K} \alpha_k^2. \]

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 \( \{\alpha_k\} \in K \rightarrow 0 \). But this contradicts the definition of \( K \) since (25) has \( \alpha_k > \rho / (L \tau) \) for all \( k \in K \).

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

**Lemma 8.** Let \( \{\phi_k\} \subset \mathbb{R}_+ \) and \( \{\chi_k\} \subset \mathbb{R}_+ \) be nonnegative scalar sequences such that, for all \( k \in \mathbb{N} \),
\[ \sum_{k=0}^{\infty} \phi_k < \infty \text{ and } [\chi_{k+1} - \chi_k]_+ \leq \phi_k. \] (26)

Then, the sequence \( \{\chi_k\} \) is convergent.

**Proof.** Let \( \Delta \chi_k^+ := \sum_{j=1}^{k}[\chi_j - \chi_j]_+ \) and \( \Delta \chi_k^- := \sum_{j=1}^{k}[\chi_j - \chi_j]_- \) 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
\[ \chi_1 + \Delta \chi_k^+ - \Delta \chi_k^- = \chi_{k+1} \geq 0 \] (27)
for all \( k \in \mathbb{N} \), the sequence \( \{\chi_k\} \) 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
\[ (E \xi_k[f(k_{n+1})] - f_{\min}) - (f(k_n) - f_{\min}) \leq \frac{1}{2} \alpha_n^2 L \sigma. \] (28)

Now taking total expectations, it follows that
\[ E[f(k_{n+1}) - f_{\min}] - E[f(k_n) - f_{\min}] \leq E[(E \xi_k[f(k_{n+1})] - f_{\min}) - (f(k_n) - f_{\min})] \leq \frac{1}{2} \alpha_n^2 L \sigma. \]

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

Now return to the result of Lemma 4, where taking total expectations and summing over the first \( K \) iterations yields
\[ E[f(k_{K+1}) - f_{\min}] - E[f(k_1) - f_{\min}] \leq - \sum_{k=1}^{K} \alpha_k (\rho - \frac{1}{2} \alpha_k L \tau) \|\nabla f(k_n)\|^2 + \frac{1}{2} L \sigma \sum_{k=1}^{K} \alpha_k^2. \]

Since the left-hand side of this expression is convergent as \( K \rightarrow \infty \), it follows that the right-hand side is bounded below as \( K \rightarrow \infty \). Then, since (15) ensures that the last sum on the right-hand side is convergent as \( K \rightarrow \infty \), it follows along with the result of Lemma 7 that
\[ \sum_{k=1}^{K} \alpha_k \|\nabla f(k_n)\|^2 \]
is convergent as \( K \rightarrow \infty \), which is (16). It then follows from (16) that (17) holds.
Lemma 4, and (29), 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

$$
E_{\xi_k} [f(w_{k+1}) - f(w_k)] \
\leq - \alpha (\rho - \frac{1}{2} \alpha L \tau) \|\nabla f(w_k)\|_2^2 + \frac{1}{2} \alpha^2 L \sigma \
\leq - \frac{1}{2} \alpha \rho \|\nabla f(w_k)\|_2^2 + \frac{1}{2} \alpha^2 L \sigma \
\leq - \alpha c \rho (f(w_k) - f(w_*)) + \frac{1}{2} \alpha^2 L \sigma.
$$

Taking total expectations, this yields

$$
E[f(w_{k+1}) - f(w_*)] \
\leq (1 - \alpha c \rho) E[f(w_k) - f(w_*)] + \frac{1}{2} \alpha^2 L \sigma.
$$

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

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

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 $k \in \mathbb{N}$,

$$
E_{\xi_k} [f(w_{k+1}) - f(w_k)] \
\leq - \alpha_k (\rho - \frac{1}{2} \alpha_k L \tau) \|\nabla f(w_k)\|_2^2 + \frac{1}{2} \alpha_k^2 L \sigma \
\leq - \frac{1}{2} \alpha_k \rho \|\nabla f(w_k)\|_2^2 + \frac{1}{2} \alpha_k^2 L \sigma \
\leq - \alpha_k c \rho (f(w_k) - f(w_*)) + \frac{1}{2} \alpha_k^2 L \sigma.
$$

Taking total expectations, this yields

$$
E[f(w_{k+1}) - f(w_*)] \
\leq (1 - \alpha_k c \rho) E[f(w_k) - f(w_*)] + \frac{1}{2} \alpha_k^2 L \sigma.
$$

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

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

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

\[\square\]