RES: Regularized Stochastic BFGS Algorithm

Aryan Mokhtari and Alejandro Ribeiro, *Member, IEEE*

***Abstract*—RES, a regularized stochastic version of the Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton method, is proposed to solve strongly convex optimization problems with stochastic objectives. The use of stochastic gradient descent algorithms is widespread, but the number of iterations required to approximate optimal arguments can be prohibitive in high dimensional problems. Application of second-order methods, on the other hand, is impracticable because the computation of objective function Hessian inverses incurs excessive computational cost. BFGS modifies gradient descent by introducing a Hessian approximation matrix computed from finite gradient differences. RES utilizes stochastic gradients in lieu of deterministic gradients for both the determination of descent directions and the approximation of the objective function's curvature. Since stochastic gradients can be computed at manageable computational cost, RES is realizable and retains the convergence rate advantages of its deterministic counterparts. Convergence results show that lower and upper bounds on the Hessian eigenvalues of the sample functions are sufficient to guarantee almost sure convergence of a subsequence generated by RES and convergence of the sequence in expectation to optimal arguments. Numerical experiments showcase reductions in convergence time relative to stochastic gradient descent algorithms and non-regularized stochastic versions of BFGS. An application of RES to the implementation of support vector machines is developed.**

***Index Terms*— Quasi-Newton methods, large-scale optimization, stochastic optimization, support vector machines.**

I. Introduction

STOCHASTIC optimization algorithms are used to solve the problem of optimizing an objective function over a set of feasible values in situations where the objective function is defined as an expectation over a set of random functions. To be precise, consider an optimization variable $\mathbf{w} \in \mathbb{R}^n$ and a random variable $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$ that determines the choice of a function $f(\mathbf{w}, \boldsymbol{\theta}) : \mathbb{R}^{n \times p} \to \mathbb{R}$. The stochastic optimization problems considered in this paper entail determination of the argument $\mathbf{w}^*$ that minimizes the expected value $F(\mathbf{w}) := \mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{w}, \boldsymbol{\theta})]$,

$$\mathbf{w}^* := \operatorname*{argmin}_{\mathbf{w}} \mathbb{E}_{\boldsymbol{\theta}} \left[f(\mathbf{w}, \boldsymbol{\theta})\right] := \operatorname*{argmin}_{\mathbf{w}} F(\mathbf{w}). \tag{1}$$

We refer to $f(\mathbf{w}, \boldsymbol{\theta})$ as the random or instantaneous functions and to $F(\mathbf{w}) := \mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{w}, \boldsymbol{\theta})]$ as the average function. We assume that the instantaneous functions $f(\mathbf{w}, \boldsymbol{\theta})$ are strongly convex for all θ from which it follows that the average function $F(\mathbf{w})$ is also strongly convex. Problems having the form in (1) are common in machine learning [3]–[5] as well as in optimal resource allocation in wireless systems [6]–[8].

Since the objective function of (1) is strongly convex, descent algorithms can be used for its minimization. However, descent methods require exact determination of the objective function's gradient $\nabla_{\mathbf{w}} F(\mathbf{w}) = \mathbb{E}_{\boldsymbol{\theta}}[\nabla_{\mathbf{w}} f(\mathbf{w}, \boldsymbol{\theta})]$, which is intractable in general. Stochastic gradient descent (SGD) methods overcome this issue by using unbiased gradient estimates based on small data samples and are the workhorse methodology used to solve large-scale stochastic optimization problems [4], [9]–[13]. Practical appeal of SGD methods remains limited, however, because they require a large number of iterations to converge. Indeed, SGD inherits slow convergence from its use of gradients which is aggravated by their replacement with stochastic estimates.

Alternatives to reduce randomness in SGD have been proposed to render the convergence times of SGD closer to the convergence times of gradient descent. Some early methods make use of memory to either smooth iterates [14] or stochastic gradients [15]. More recent developments have focused on hybrid approaches that use both, gradients and stochastic gradients, or update descent directions so that they become progressively closer to gradients [13], [16], [17]. Inasmuch as they succeed in reducing randomness, these algorithms end up exhibiting the asymptotic convergence rate of gradient descent which is faster than the asymptotic convergence rate of SGD. Although they improve asymptotic convergence rates, the latter methods are still often slow in practice. This is not unexpected. Reducing randomness is of no use when the function $F(\mathbf{w})$ has a challenging curvature profile. In these ill-conditioned functions SGD is limited by the slow convergence times of (deterministic) gradient descent.

To overcome problems with the objective function's curvature, one may think of developing stochastic versions of Newton's method. However, computing unbiased estimates of Newton steps is not easy except in problems with some specific structures [18], [19]. Recourse to quasi-Newton methods then arises as a natural alternative because they can achieve superlinear convergence rates in deterministic settings while relying on gradients to compute curvature estimates [20]–[23]. Considering the fact that unbiased gradient estimates are computable at manageable cost, stochastic generalizations of quasi-Newton

Manuscript received December 11, 2013; revised June 04, 2014; accepted August 25, 2014. Date of publication September 16, 2014; date of current version November 04, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gesualdo Scutari. Work in this paper is supported by ARO W911NF-10-1-0388, NSF CAREER CCF-0952867, and ONR N00014-12-1-0997. Part of the results in this paper were presented at the IEEE Global Conference on Signal Information Processing, Austin, TX, USA, December 3–5, 2013 [1] and the IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, May 4–9, 2014 [2].

The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: aryanm@seas.upenn.edu; aribeiro@seas.upenn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2014.2357775

methods are not difficult to devise [6], [24], [25]. Numerical tests of these methods on simple quadratic objectives suggest that stochastic quasi-Newton methods retain the convergence rate advantages of their deterministic counterparts [24]. The success of these preliminary experiments notwithstanding, Hessian estimations based on random stochastic gradients may result in near singular curvature estimates. The possibility of having singular curvature estimates makes it impossible to provide convergence analyses for stochastic quasi-Newton methods [24], [25] and may result in erratic numerical behavior (see Section V-B).

In this paper we introduce a stochastic regularized version of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method to solve problems with the generic structure in (1). The proposed regularization avoids the near-singularity problems of more straightforward extensions and yields an algorithm with provable convergence guarantees when the functions $f(\mathbf{w},\boldsymbol{\theta})$ are strongly convex. We begin the paper with a brief discussion of SGD (Section II) and deterministic BFGS (Section II-A). The fundamental idea of BFGS is to continuously satisfy a secant condition that captures information on the curvature of the function being minimized while staying close to previous curvature estimates. To regularize deterministic BFGS we retain the secant condition but modify the proximity condition so that the eigenvalues of the Hessian approximation matrix stay above a given threshold (Section II-A). This regularized version is leveraged to introduce the regularized stochastic BFGS algorithm (Section II-B). Regularized stochastic BFGS differs from standard BFGS in the use of regularization to make a bound on the largest eigenvalue of the Hessian inverse approximation matrix and in the use of stochastic gradients in lieu of deterministic gradients for both the determination of descent directions and the approximation of the objective function's curvature. We abbreviate regularized stochastic BFGS as RES.

Convergence properties of RES are then analyzed (Section III). We prove that lower and upper bounds on the Hessians of the sample functions $f(\mathbf{w},\boldsymbol{\theta})$ are sufficient to guarantee convergence of a subsequence to the optimal argument $\mathbf{w}^*$ with probability 1 over realizations of the sample functions (Theorem 1). We complement this result with a characterization of the convergence rate which is shown to be at least of order $O(1/t)$ in expectation (Theorem 2). This expected convergence rate is typical of stochastic optimization algorithms and, in that sense, no better than SGD [11]. Advantages of RES relative to SGD are nevertheless significant, as we establish in numerical results for the minimization of a family of quadratic objective functions of varying dimensionality and condition number (Section IV). As we vary the condition number we observe that for well-conditioned objectives RES and SGD exhibit comparable performance, whereas for ill-conditioned functions RES outperforms SGD by an order of magnitude (Section IV-A). As we vary problem dimension we observe that SGD becomes unworkable for large-dimensional problems. RES however, exhibits manageable degradation as the number of iterations required for convergence doubles when the problem dimension increases by a factor of ten (Section IV-D).

An important example of a class of problems having the form in (1) are support vector machines (SVMs) that reduce binary classification to the determination of a hyperplane that separates points in a given training set; see, e.g., [4], [26], [27]. We adapt RES for SVM problems (Section V) and show the improvement relative to SGD in convergence time and stability through numerical analysis (Section V-A). For this particular problem of finding optimal SVM classifiers, several accelerations of SGD have been proposed. These include the Stochastic Average Gradient (SAG) method [13], the Semi-Stochastic Gradient Descent (S2GD) algorithm [16], and Stochastic Approximation by Averaging (SAA) [14]. The comparison of RES with these accelerated versions yields the expected conclusion. SAG and S2GD accelerate the convergence of SGD but still underperform RES for problems that are not well-conditioned. As we commented above, RES solves a different problem than the one targeted by SAG, S2GD, and SAA. The latter attempt to reduce the randomness in SGD to make the convergence rate closer to that of gradient descent. RES attempts to adapt to the curvature of the objective function. We also compare RES to standard (non-regularized) stochastic BFGS. The regularization in RES is fundamental in guaranteeing convergence as standard (non-regularized) stochastic BFGS is observed to routinely fail in the computation of a separating hyperplane.

Notation: Lowercase boldface $\mathbf{v}$ denotes a vector and uppercase boldface $\mathbf{A}$ a matrix. We use $\|\mathbf{v}\|$ to denote the Euclidean norm of vector $\mathbf{v}$ and $\|\mathbf{A}\|$ to denote the Euclidean norm of matrix $\mathbf{A}$. The trace of $\mathbf{A}$ is written as $\mathrm{tr}(\mathbf{A})$ and the determinant as $\det(\mathbf{A})$. We use $\mathbf{I}$ for the identity matrix of appropriate dimension. The notation $\mathbf{A}\succeq\mathbf{B}$ implies that the matrix $\mathbf{A}-\mathbf{B}$ is positive semidefinite. The operator $\mathbb{E}_{\mathbf{x}}[\cdot]$ stands for expectation over random variable $\mathbf{x}$ and $\mathbb{E}[\cdot]$ for expectation with respect to the distribution of a stochastic process.

II. Algorithm Definition

Recall the definitions of the sample functions $f(\mathbf{w},\boldsymbol{\theta})$ and the average function $F(\mathbf{w}):=\mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{w},\boldsymbol{\theta})]$. Since the function $F(\mathbf{w})$ is strongly convex, we can find the optimal argument $\mathbf{w}^*$ in (1) with a gradient descent algorithm. Considering that strongly convex functions are continuously differentiable and further assuming that the instantaneous functions $f(\mathbf{w},\boldsymbol{\theta})$ have finite gradients it follows that the gradients of $F(\mathbf{w})$ are given by

$$\mathbf{s}(\mathbf{w}):=\nabla F(\mathbf{w})=\mathbb{E}_{\boldsymbol{\theta}}\left[\nabla f(\mathbf{w},\boldsymbol{\theta})\right]. \tag{2}$$

When the number of functions $f(\mathbf{w},\boldsymbol{\theta})$ is large, as is the case in most problems of practical interest, exact evaluation of the gradient $\mathbf{s}(\mathbf{w})$ is impractical. This motivates the use of stochastic gradients in lieu of actual gradients. More precisely, consider a given set of L realizations $\tilde{\boldsymbol{\theta}}=[\boldsymbol{\theta}_1;\ldots;\boldsymbol{\theta}_L]$ and define the stochastic gradient of $F(\mathbf{w})$ at $\mathbf{w}$ given samples $\tilde{\boldsymbol{\theta}}$ as

$$\hat{\mathbf{s}}(\mathbf{w},\tilde{\boldsymbol{\theta}}):=\frac{1}{L}\sum_{l=1}^{L}\nabla f(\mathbf{w},\boldsymbol{\theta}_l). \tag{3}$$

Introducing now a time index t, an initial iterate $\mathbf{w}_0$, and a step size sequence ϵ_t, a stochastic gradient descent algorithm is defined by the iteration

$$\mathbf{w}_{t+1}=\mathbf{w}_t-\epsilon_t\hat{\mathbf{s}}(\mathbf{w}_t,\tilde{\boldsymbol{\theta}}_t). \tag{4}$$

To implement (4) we compute stochastic gradients $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ using (3). In turn, this requires determination of the gradients of the random functions $f(\mathbf{w}, \boldsymbol{\theta}_{tl})$ for each $\boldsymbol{\theta}_{tl}$ component of $\tilde{\boldsymbol{\theta}}_t$ and their corresponding average. The computational cost is manageable for small values of L.

The stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ in (3) is an unbiased estimate of the (average) gradient $\mathbf{s}(\mathbf{w})$ in (2) in the sense that $\mathbb{E}_{\tilde{\boldsymbol{\theta}}}[\hat{\mathbf{s}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})] = \mathbf{s}(\mathbf{w})$. Thus, the iteration in (4) is such that, on average, iterates descend along a negative gradient direction, see, e.g., [11]. This intuitive observation can be formalized into a proof of convergence when the step size sequence is selected as nonsummable but square summable, i.e.,

$$\sum_{t=0}^{\infty} \epsilon_t = \infty, \quad \text{and} \quad \sum_{t=0}^{\infty} \epsilon_t^2 < \infty. \tag{5}$$

A customary step size choice for which (5) holds is to make $\epsilon_t = \epsilon_0 T_0/(T_0 + t)$, for given parameters ϵ_0 and T_0 that control the initial step size and its speed of decrease, respectively. Convergence notwithstanding, the number of iterations required to approximate $\mathbf{w}^*$ is very large in problems that don't have small condition numbers [12]. This motivates the alternative methods we discuss in subsequent sections.

A. Regularized BFGS

To speed up convergence of (4) resorting to second order methods is of little use because evaluating Hessians of the objective function is computationally intensive. A better suited methodology is the use of quasi-Newton methods whereby gradient descent directions are premultiplied by a matrix $\mathbf{B}_t^{-1}$,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \epsilon_t \mathbf{B}_t^{-1} \mathbf{s}(\mathbf{w}_t). \tag{6}$$

The idea is to select positive definite matrices $\mathbf{B}_t \succ 0$ close to the Hessian of the objective function $\mathbf{H}(\mathbf{w}_t) := \nabla^2 F(\mathbf{w}_t)$. Various methods are known to select matrices $\mathbf{B}_t$, including those by Broyden e.g., [28]; Davidon, Fletcher, and Powell (DFP) [29]; and Broyden, Fletcher, Goldfarb, and Shanno (BFGS) e.g., [21]–[23]. We work here with the matrices $\mathbf{B}_t$ used in BFGS since they have been observed to work best in practice [22].

In BFGS—and all other quasi-Newton methods—the function's curvature is approximated by a finite difference. Specifically, define the variable and gradient variations at time t as

$$\mathbf{v}_t := \mathbf{w}_{t+1} - \mathbf{w}_t, \quad \text{and} \quad \mathbf{r}_t := \mathbf{s}(\mathbf{w}_{t+1}) - \mathbf{s}(\mathbf{w}_t), \tag{7}$$

respectively, and select the matrix $\mathbf{B}_{t+1}$ to be used in the next time step so that it satisfies the secant condition $\mathbf{B}_{t+1}\mathbf{v}_t = \mathbf{r}_t$. The rationale for this selection is that the Hessian $\mathbf{H}(\mathbf{w}_t)$ satisfies this condition for $\mathbf{w}_{t+1}$ tending to $\mathbf{w}_t$. Notice however that the secant condition $\mathbf{B}_{t+1}\mathbf{v}_t = \mathbf{r}_t$ is not enough to completely specify $\mathbf{B}_{t+1}$. To resolve this indeterminacy, matrices $\mathbf{B}_{t+1}$ in BFGS are also required to be as close as possible to $\mathbf{B}_t$ in terms of the Gaussian differential entropy,

$$\begin{aligned} \mathbf{B}_{t+1} = \underset{\mathbf{Z}}{\operatorname{argmin}} \quad & \operatorname{tr}\left[\mathbf{B}_t^{-1}\mathbf{Z}\right] - \log\det\left[\mathbf{B}_t^{-1}\mathbf{Z}\right] - n, \\ \text{s.t.} \quad & \mathbf{Z}\mathbf{v}_t = \mathbf{r}_t, \quad \mathbf{Z} \succeq \mathbf{0}. \end{aligned} \tag{8}$$

The constraint $\mathbf{Z} \succeq \mathbf{0}$ in (8) restricts the feasible space to positive semidefinite matrices whereas the constraint $\mathbf{Z}\mathbf{v}_t = \mathbf{r}_t$ requires $\mathbf{Z}$ to satisfy the secant condition. The objective $\operatorname{tr}(\mathbf{B}_t^{-1}\mathbf{Z}) - \log\det(\mathbf{B}_t^{-1}\mathbf{Z}) - n$ represents the differential entropy between random variables with zero-mean Gaussian distributions $\mathcal{N}(\mathbf{0}, \mathbf{B}_t)$ and $\mathcal{N}(\mathbf{0}, \mathbf{Z})$ having covariance matrices $\mathbf{B}_t$ and $\mathbf{Z}$. The differential entropy is nonnegative and equal to zero if and only if $\mathbf{Z} = \mathbf{B}_t$. The solution $\mathbf{B}_{t+1}$ of the semidefinite program in (8) is therefore closest to $\mathbf{B}_t$ in the sense of minimizing the Gaussian differential entropy among all positive semidefinite matrices that satisfy the secant condition $\mathbf{Z}\mathbf{v}_t = \mathbf{r}_t$.

Strongly convex functions are such that the inner product of the gradient and variable variations is positive, i.e., $\mathbf{v}_t^T \mathbf{r}_t > 0$. In that case the matrix $\mathbf{B}_{t+1}$ in (8) is explicitly given by the update—see, e.g., [23] and the proof of Lemma 1—,

$$\mathbf{B}_{t+1} = \mathbf{B}_t + \frac{\mathbf{r}_t\mathbf{r}_t^T}{\mathbf{v}_t^T\mathbf{r}_t} - \frac{\mathbf{B}_t\mathbf{v}_t\mathbf{v}_t^T\mathbf{B}_t}{\mathbf{v}_t^T\mathbf{B}_t\mathbf{v}_t}. \tag{9}$$

In principle, the solution to (8) could be positive semidefinite but not positive definite, i.e., we can have $\mathbf{B}_{t+1} \succeq \mathbf{0}$ but $\mathbf{B}_{t+1} \nsucc \mathbf{0}$. However, through direct operation in (9) it is not difficult to conclude that $\mathbf{B}_{t+1}$ stays positive definite if the matrix $\mathbf{B}_t$ is positive definite. Thus, initializing the curvature estimate with a positive definite matrix $\mathbf{B}_0 \succ \mathbf{0}$ guarantees $\mathbf{B}_t \succ \mathbf{0}$ for all subsequent times t. Still, it is possible for the smallest eigenvalue of $\mathbf{B}_t$ to become arbitrarily close to zero which means that the largest eigenvalue of $\mathbf{B}_t^{-1}$ can become arbitrarily large. This has been proven not to be an issue in BFGS implementations but is a significant challenge in the stochastic version proposed here.

To avoid this problem we introduce a regularization of (8) to enforce the eigenvalues of $\mathbf{B}_{t+1}$ to exceed a positive constant δ. Specifically, we redefine $\mathbf{B}_{t+1}$ as the solution of problem,

$$\begin{aligned} \mathbf{B}_{t+1} = \underset{\mathbf{Z}}{\operatorname{argmin}} \quad & \operatorname{tr}\left[\mathbf{B}_t^{-1}(\mathbf{Z}-\delta\mathbf{I})\right] - \log\det\left[\mathbf{B}_t^{-1}(\mathbf{Z}-\delta\mathbf{I})\right] - n, \\ \text{s.t.} \quad & \mathbf{Z}\mathbf{v}_t = \mathbf{r}_t, \quad \mathbf{Z} \succeq \mathbf{0}. \end{aligned} \tag{10}$$

The curvature approximation matrix $\mathbf{B}_{t+1}$ defined in (10) still satisfies the secant condition $\mathbf{B}_{t+1}\mathbf{v}_t = \mathbf{r}_t$ but has a different proximity requirement since instead of comparing $\mathbf{B}_t$ and $\mathbf{Z}$ we compare $\mathbf{B}_t$ and $\mathbf{Z} - \delta\mathbf{I}$. While (10) does not ensure that all eigenvalues of $\mathbf{B}_{t+1}$ exceed δ we can show that this will be the case under two minimally restrictive assumptions. We do so in the following proposition where we also give an explicit solution for (10) analogous to the expression in (9) that solves the non-regularized problem in (8).

Proposition 1: Consider the semidefinite program in (10) where the matrix $\mathbf{B}_t \succ \mathbf{0}$ is positive definite and define the corrected gradient variation

$$\tilde{\mathbf{r}}_t := \mathbf{r}_t - \delta\mathbf{v}_t, \tag{11}$$

where $\delta > 0$ is a constant. If the inner product $\tilde{\mathbf{r}}_t^T\mathbf{v}_t = (\mathbf{r}_t - \delta\mathbf{v}_t)^T\mathbf{v}_t$ is positive, the solution $\mathbf{B}_{t+1}$ of (10) is such that all eigenvalues of $\mathbf{B}_{t+1}$ are larger than δ,

$$\mathbf{B}_{t+1} \succeq \delta\mathbf{I}. \tag{12}$$

Furthermore, $\mathbf{B}_{t+1}$ is explicitly given by the expression

$$\mathbf{B}_{t+1} = \mathbf{B}_t + \frac{\tilde{\mathbf{r}}_t \tilde{\mathbf{r}}_t^T}{\mathbf{v}_t^T \tilde{\mathbf{r}}_t} - \frac{\mathbf{B}_t \mathbf{v}_t \mathbf{v}_t^T \mathbf{B}_t}{\mathbf{v}_t^T \mathbf{B}_t \mathbf{v}_t} + \delta \mathbf{I}. \tag{13}$$

Proof: See Appendix. ■

Comparing (9) and (13), it follows that the differences between BFGS and regularized BFGS are the replacement of the gradient variation $\mathbf{r}_t$ in (7) by the corrected variation $\tilde{\mathbf{r}}_t := \mathbf{r}_t - \delta \mathbf{v}_t$ and the addition of the regularization term $\delta \mathbf{I}$. We use (13) in the construction of the stochastic BFGS in the following section.

B. RES: Regularized Stochastic BFGS

As can be seen from (13) the regularized BFGS curvature estimate $\mathbf{B}_{t+1}$ is obtained as a function of previous estimates $\mathbf{B}_t$, iterates $\mathbf{w}_t$ and $\mathbf{w}_{t+1}$, and corresponding gradients $\mathbf{s}(\mathbf{w}_t)$ and $\mathbf{s}(\mathbf{w}_{t+1})$. We can then think of a method in which gradients $\mathbf{s}(\mathbf{w}_t)$ are replaced by stochastic gradients $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ in both the curvature approximation update in (13) and the descent iteration in (6). Specifically, start at time t with current iterate $\mathbf{w}_t$ and let $\mathbf{B}_t$ stand for the Hessian approximation computed by stochastic BFGS in the previous iteration. Obtain a batch of samples $\tilde{\boldsymbol{\theta}}_t = [\boldsymbol{\theta}_{t1}; \ldots; \boldsymbol{\theta}_{tL}]$, determine the value of the stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ as per (3), and update the iterate $\mathbf{w}_t$ as

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \epsilon_t \left(\hat{\mathbf{B}}_t^{-1} + \Gamma \mathbf{I} \right) \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t), \tag{14}$$

where we added the identity bias term $\Gamma \mathbf{I}$ for a given positive constant $\Gamma > 0$. Relative to SGD as defined by (4), RES as defined by (14) differs in the use of the matrix $\hat{\mathbf{B}}_t^{-1} + \Gamma \mathbf{I}$ to account for the curvature of $F(\mathbf{w})$. Relative to (regularized or non-regularized) BFGS as defined in (6) RES differs in the use of stochastic gradients $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ instead of actual gradients and in the use of the curvature approximation $\hat{\mathbf{B}}_t^{-1} + \Gamma \mathbf{I}$ in lieu of $\mathbf{B}_t^{-1}$. Observe that in (14) we add a bias $\Gamma \mathbf{I}$ to the curvature approximation $\hat{\mathbf{B}}_t^{-1}$. This is necessary to ensure convergence by hedging against random variations in $\hat{\mathbf{B}}_t^{-1}$ as we discuss in Section III.

To update the Hessian approximation matrix $\hat{\mathbf{B}}_t$ compute the stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t)$ associated with the *same* set of samples $\tilde{\boldsymbol{\theta}}_t$ used to compute the stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$. Define then the stochastic gradient variation at time t as

$$\hat{\mathbf{r}}_t := \hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t), \tag{15}$$

and redefine $\tilde{\mathbf{r}}_t$ so that it stands for the modified stochastic gradient variation

$$\tilde{\mathbf{r}}_t := \hat{\mathbf{r}}_t - \delta \mathbf{v}_t, \tag{16}$$

by using $\hat{\mathbf{r}}_t$ instead of $\mathbf{r}_t$. The Hessian approximation $\hat{\mathbf{B}}_{t+1}$ for the next iteration is defined as the matrix that satisfies the stochastic secant condition $\mathbf{Z}\mathbf{v}_t = \hat{\mathbf{r}}_t$ and is closest to $\hat{\mathbf{B}}_t$ in the sense of (10). As per Proposition 1 we can compute $\hat{\mathbf{B}}_{t+1}$ explicitly as

$$\hat{\mathbf{B}}_{t+1} = \hat{\mathbf{B}}_t + \frac{\tilde{\mathbf{r}}_t \tilde{\mathbf{r}}_t^T}{\mathbf{v}_t^T \tilde{\mathbf{r}}_t} - \frac{\hat{\mathbf{B}}_t \mathbf{v}_t \mathbf{v}_t^T \hat{\mathbf{B}}_t}{\mathbf{v}_t^T \hat{\mathbf{B}}_t \mathbf{v}_t} + \delta \mathbf{I}. \tag{17}$$

as long as $(\hat{\mathbf{r}}_t - \delta \mathbf{v}_t)^T \mathbf{v}_t = \tilde{\mathbf{r}}^T \mathbf{v}_t > 0$. Conditions to guarantee that $\tilde{\mathbf{r}}_t^T \mathbf{v}_t > 0$ are introduced in Section III.

Algorithm 1: RES: Regularized Stochastic BFGS

```
Require: Variable $\mathbf{w}_0$. Hessian approximation $\hat{\mathbf{B}}_0 \succ \delta\mathbf{I}$.
1: for $t = 0, 1, 2, \ldots$ do
2:    Acquire $L$ independent samples $\tilde{\boldsymbol{\theta}}_t = [\boldsymbol{\theta}_{t1}, \ldots, \boldsymbol{\theta}_{tL}]$
3:    Compute $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ [cf. (3)]
          $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t) = \frac{1}{L} \sum_{l=1}^{L} \nabla_{\mathbf{w}} f(\mathbf{w}_t, \boldsymbol{\theta}_{tl}).$
4:    Descend along direction $(\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I})\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ [cf. (14)]
          $\mathbf{w}_{t+1} = \mathbf{w}_t - \epsilon_t \left(\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}\right) \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t).$
5:    Compute $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t)$ [cf. (3)]
          $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) = \frac{1}{L} \sum_{l=1}^{L} \nabla_{\mathbf{w}} f(\mathbf{w}_{t+1}, \boldsymbol{\theta}_{tl}).$
6:    Compute variable variation [cf. (7)] $\mathbf{v}_t = \mathbf{w}_{t+1} - \mathbf{w}_t$.
7:    Compute modified stochastic gradient variation [cf. (16)]
          $\tilde{\mathbf{r}}_t = \hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t) - \delta\mathbf{v}_t.$
8:    Update Hessian approximation matrix [cf. (17)]
          $\hat{\mathbf{B}}_{t+1} = \hat{\mathbf{B}}_t + \frac{\tilde{\mathbf{r}}_t \tilde{\mathbf{r}}_t^T}{\mathbf{v}_t^T \tilde{\mathbf{r}}_t} - \frac{\hat{\mathbf{B}}_t \mathbf{v}_t \mathbf{v}_t^T \hat{\mathbf{B}}_t}{\mathbf{v}_t^T \hat{\mathbf{B}}_t \mathbf{v}_t} + \delta\mathbf{I}.$
9: end for
```

The resulting RES algorithm is summarized in Algorithm 1. The two core steps in each iteration are the descent in Step 4 and the update of the Hessian approximation $\hat{\mathbf{B}}_t$ in Step 8. Step 2 comprises the observation of L samples that are required to compute the stochastic gradients in Steps 3 and 5. The stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ in Step 3 is used in the descent iteration in Step 4. The stochastic gradient of Step 3 along with the stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t)$ of Step 5 are used to compute the variations in Steps 6 and 7 that permit carrying out the update of the Hessian approximation $\hat{\mathbf{B}}_t$ in Step 8. Iterations are initialized at arbitrary variable $\mathbf{w}_0$ and positive definite matrix $\hat{\mathbf{B}}_0$ with the smallest eigenvalue larger than δ.

Remark 1: One may think that the natural substitution of the gradient variation $\mathbf{r}_t = \mathbf{s}(\mathbf{w}_{t+1}) - \mathbf{s}(\mathbf{w}_t)$ is the stochastic gradient variation $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_{t+1}) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ instead of the variation $\hat{\mathbf{r}}_t = \hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ in (15). This would have the advantage that $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_{t+1})$ is the stochastic gradient used to descend in iteration $t+1$ whereas $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t)$ is not and is just computed for the purposes of updating $\mathbf{B}_t$. Therefore, using the variation $\hat{\mathbf{r}}_t = \hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ requires twice as many stochastic gradient evaluations as using the variation $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_{t+1}) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$. However, the use of the variation $\hat{\mathbf{r}}_t = \hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ is necessary to ensure that $(\hat{\mathbf{r}}_t - \delta \mathbf{v}_t)^T \mathbf{v}_t = \tilde{\mathbf{r}}_t^T \mathbf{v}_t > 0$, which in turn is required for (17) to be true. This cannot be guaranteed if we use the variation $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_{t+1}) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$—see Lemma 1 for details. The same observation holds true for the non-regularized version of stochastic BFGS introduced in [24].

III. Convergence

For the subsequent analysis we define the instantaneous objective function associated with samples $\tilde{\boldsymbol{\theta}} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L]$ as

$$\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}}) := \frac{1}{L}\sum_{l=1}^{L} f(\mathbf{w}, \boldsymbol{\theta}_l). \tag{18}$$

The definition of the instantaneous objective function $\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ in association with the fact that $F(\mathbf{w}) := \mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{w}, \boldsymbol{\theta})]$ implies

$$F(\mathbf{w}) = \mathbb{E}_{\boldsymbol{\theta}}\left[\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}})\right]. \tag{19}$$

Our goal here is to show that as time progresses the sequence of variable iterates $\mathbf{w}_t$ approaches the optimal argument $\mathbf{w}^*$. In proving this result we make the following assumptions.

Assumption 1: The instantaneous functions $\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ are twice differentiable and the eigenvalues of the instantaneous Hessian $\hat{\mathbf{H}}(\mathbf{w}, \tilde{\boldsymbol{\theta}}) = \nabla^2_{\mathbf{w}} \hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ are bounded between constants $0 < \tilde{m}$ and $\tilde{M} < \infty$ for all random variables $\tilde{\boldsymbol{\theta}}$,

$$\tilde{m}\mathbf{I} \preceq \hat{\mathbf{H}}(\mathbf{w}, \tilde{\boldsymbol{\theta}}) \preceq \tilde{M}\mathbf{I}. \tag{20}$$

Assumption 2: The second moment of the norm of the stochastic gradient is bounded for all $\mathbf{w}$, i.e., there exists a constant S^2 such that for all variables $\mathbf{w}$, it holds

$$\mathbb{E}_{\boldsymbol{\theta}}\left[\left\|\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)\right\|^2 |\mathbf{w}_t\right] \leq S^2. \tag{21}$$

Assumption 3: The regularization constant δ is smaller than the smallest Hessian eigenvalue $\tilde{m}$, i.e., $\delta < \tilde{m}$.

As a consequence of Assumption 1 similar eigenvalue bounds hold for the (average) function $F(\mathbf{w})$. Indeed, it follows from the linearity of the expectation operator and the expression in (19) that the Hessian is $\nabla^2_{\mathbf{w}} F(\mathbf{w}) = \mathbf{H}(\mathbf{w}) = \mathbb{E}_{\boldsymbol{\theta}}[\hat{\mathbf{H}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})]$. Combining this observation with the bounds in (20) it follows that there are constants $m \geq \tilde{m}$ and $M \leq \tilde{M}$ such that

$$\tilde{m}\mathbf{I} \preceq m\mathbf{I} \preceq \mathbf{H}(\mathbf{w}) \preceq M\mathbf{I} \preceq \tilde{M}\mathbf{I}. \tag{22}$$

The bounds in (22) are customary in convergence proofs of descent methods. For the results here the stronger condition spelled in Assumption 1 is needed. The lower bound implies strong convexity of the instantaneous functions and the upper bound is equivalent to them having Lipschitz Continuous gradients. The restriction imposed by Assumption 2 is typical of stochastic descent algorithms, its intent being to limit the random variation of stochastic gradients [11]. Assumption 3 is necessary to guarantee that the inner product $\tilde{\mathbf{r}}_t^T \mathbf{v}_t = (\mathbf{r}_t - \delta \mathbf{v}_t)^T \mathbf{v}_t > 0$ [cf. Proposition 1] is positive as we show in the following lemma.

Lemma 1: Consider the modified stochastic gradient variation $\tilde{\mathbf{r}}_t$ defined in (16) and the variable variation $\mathbf{v}_t$ defined in (7). Let Assumption 1 hold and recall the lower bound $\tilde{m}$ on the smallest eigenvalue of the instantaneous Hessians. Then, for all constants $0 < \delta < \tilde{m}$, it holds

$$\tilde{\mathbf{r}}_t^T \mathbf{v}_t = (\hat{\mathbf{r}}_t - \delta \mathbf{v}_t)^T \mathbf{v}_t \geq (\tilde{m} - \delta)\|\mathbf{v}_t\|^2 > 0. \tag{23}$$

Proof: As per (20) in Assumption 1 the eigenvalues of the instantaneous Hessian $\hat{\mathbf{H}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ are bounded below by $\tilde{m} > 0$ which is equivalent to saying that instantaneous objective functions $\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ associated with samples $\tilde{\boldsymbol{\theta}}$ are m-strongly convex with respect to $\mathbf{w}$. Considering the strong monotonicity of gradients for the m-strongly convex functions $\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}}_t)$, we can write

$$\left[\nabla\hat{f}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) - \nabla\hat{f}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)\right]^T (\mathbf{w}_{t+1} - \mathbf{w}_t) \geq \tilde{m}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2. \tag{24}$$

Observing the definitions of stochastic gradients $\hat{\mathbf{s}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ in (3) and instantaneous objective functions $\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ in (18) it follows that $\nabla\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}}) = \hat{\mathbf{s}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$. Hence, we can rewrite (24) as

$$\left(\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)\right)^T (\mathbf{w}_{t+1} - \mathbf{w}_t) \geq \tilde{m}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2. \tag{25}$$

Using the definitions of stochastic gradient variation $\hat{\mathbf{r}}_t$ and variable variation $\mathbf{v}_t$ in (15) and (7) we further simplify (25) to

$$\hat{\mathbf{r}}_t^T \mathbf{v}_t \geq \tilde{m}\|\mathbf{v}_t\|^2. \tag{26}$$

Consider now the inner product $\tilde{\mathbf{r}}_t^T \mathbf{v}_t = (\hat{\mathbf{r}}_t - \delta \mathbf{v}_t)^T \mathbf{v}_t$ in (23) and use the bound in (26) to write

$$\tilde{\mathbf{r}}_t^T \mathbf{v}_t = \hat{\mathbf{r}}_t^T \mathbf{v}_t - \delta \mathbf{v}_t^T \mathbf{v}_t \geq (\tilde{m} - \delta)\|\mathbf{v}_t\|^2.$$

Since we are selecting $\delta < \tilde{m}$ by hypothesis it follows that (23) is true for all times t. ■

Initializing the curvature approximation matrix as $\hat{\mathbf{B}}_0 \succ \delta\mathbf{I}$, which implies $\hat{\mathbf{B}}_0^{-1} \succ \mathbf{0}$, and setting $\delta < \tilde{m}$ it follows from Lemma 1 that the hypotheses of Proposition 1 are satisfied for $t = 0$. Hence, the matrix $\hat{\mathbf{B}}_1$ computed from (17) is the solution of the semidefinite program in (10) and, more to the point, satisfies $\hat{\mathbf{B}}_1 \succ \delta\mathbf{I}$, which in turn implies $\hat{\mathbf{B}}_1^{-1} \succ \mathbf{0}$. Proceeding recursively we can conclude that $\hat{\mathbf{B}}_t \succ \delta\mathbf{I} \succ \mathbf{0}$ for all times $t \geq 0$. Equivalently, this implies that all the eigenvalues of $\hat{\mathbf{B}}_t^{-1}$ are between 0 and $1/\delta$ and that, as a consequence, the matrix $\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}$ is such that

$$\Gamma\mathbf{I} \preceq \hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I} \preceq \left(\Gamma + \left(\frac{1}{\delta}\right)\right)\mathbf{I}. \tag{27}$$

Having matrices $\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}$ that are strictly positive definite with eigenvalues uniformly upper bounded by $\Gamma + (1/\delta)$ leads to the conclusion that if $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ is a descent direction, the same holds true of $(\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I})\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$. The stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ is not a descent direction in general, but we know that this is true for its conditional expectation $\mathbb{E}[\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)|\mathbf{w}_t] = \nabla_{\mathbf{w}} F(\mathbf{w}_t)$. Therefore, we conclude that $(\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I})\hat{\mathbf{s}}(\mathbf{w}_t, \boldsymbol{\theta}_t)$ is an average descent direction because $\mathbb{E}[(\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I})\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)|\mathbf{w}_t] = (\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I})\nabla_{\mathbf{w}} F(\mathbf{w}_t)$. Stochastic optimization algorithms whose displacements $\mathbf{w}_{t+1} - \mathbf{w}_t$ are descent directions on average are expected to approach optimal arguments in a sense that we specify formally in the following lemma.

Lemma 2: Consider the RES algorithm as defined by (14)–(17). If Assumptions 1, 2, and 3 hold true, the sequence of average function $F(\mathbf{w}_t)$ satisfies

$$\mathbb{E}\left[F(\mathbf{w}_{t+1})|\mathbf{w}_t\right] \leq F(\mathbf{w}_t) - \epsilon_t\Gamma \left\|\nabla F(\mathbf{w}_t)\right\|^2 + K\epsilon_t^2 \quad (28)$$

where the constant $K := MS^2(1/\delta + \Gamma)^2/2$.

Proof: It follows from Assumption 1 that the eigenvalues of the Hessian $\mathbf{H}(\mathbf{w}_t) = \mathbb{E}_{\tilde{\boldsymbol{\theta}}}[\hat{\mathbf{H}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)] = \nabla^2_{\mathbf{w}}F(\mathbf{w}_t)$ are bounded between $0 < m$ and $M < \infty$ as stated in (22). Taking a Taylor's expansion of the dual function $F(\mathbf{w})$ around $\mathbf{w} = \mathbf{w}_t$ and using the upper bound in the Hessian eigenvalues we can write

$$F(\mathbf{w}_{t+1}) \leq F(\mathbf{w}_t) + \nabla F(\mathbf{w}_t)^T(\mathbf{w}_{t+1} - \mathbf{w}_t) + \frac{M}{2}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2 \quad (29)$$

From the definition of the RES update in (14) we can write the difference of two consecutive variables $\mathbf{w}_{t+1} - \mathbf{w}_t$ as $-\epsilon_t(\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I})\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$. Making this substitution in (29), taking expectation with $\mathbf{w}_t$ given in both sides of the resulting inequality, and observing the fact that when $\mathbf{w}_t$ is given the Hessian approximation $\hat{\mathbf{B}}_t^{-1}$ is deterministic we can write

$$\begin{aligned}&\mathbb{E}[F(\mathbf{w}_{t+1})|\mathbf{w}_t]\\ &\quad\leq F(\mathbf{w}_t) - \epsilon_t\nabla F(\mathbf{w}_t)^T\left(\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}\right)\mathbb{E}\left[\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)|\mathbf{w}_t\right]\\ &\qquad+\frac{\epsilon^2 M}{2}\mathbb{E}\left[\left\|\left(\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}\right)\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)\right\|^2|\mathbf{w}_t\right]. \quad (30)\end{aligned}$$

We proceed to bound the third term in the right hand side of (30). Start by observing that the 2-norm of a product is not larger than the product of the 2-norms and that, as noted above, with $\mathbf{w}_t$ given the matrix $\hat{\mathbf{B}}_t^{-1}$ is also given to write

$$\begin{aligned}&\mathbb{E}\left[\left\|\left(\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}\right)\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)\right\|^2|\mathbf{w}_t\right]\\ &\qquad\leq \left\|\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}\right\|^2\mathbb{E}\left[\left\|\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)\right\|^2|\mathbf{w}_t\right]. \quad (31)\end{aligned}$$

Notice that, as stated in (27), $\Gamma + 1/\delta$ is an upper bound for the eigenvalues of $\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}$. Further observe that the second moment of the norm of the stochastic gradient is bounded by $\mathbb{E}[\|\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)\|^2|\mathbf{w}_t] \leq S^2$, as stated in Assumption 2. These two upper bounds substituted in (31) yield

$$\mathbb{E}\left[\left\|\left(\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}\right)\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)\right\|^2|\mathbf{w}_t\right] \leq S^2(1/\delta + \Gamma)^2. \quad (32)$$

Substituting the upper bound in (32) for the third term of (30) and further using the fact that $\mathbb{E}[\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)|\mathbf{w}_t] = \nabla F(\mathbf{w}_t)$ in the second term leads to

$$\begin{aligned}\mathbb{E}\left[F(\mathbf{w}_{t+1})|\mathbf{w}_t\right] \leq F(\mathbf{w}_t) - \epsilon_t\nabla F(\mathbf{w}_t)^T\left[\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}\right]\nabla F(\mathbf{w}_t)\\ +\frac{\epsilon_t^2 MS^2}{2}(1/\delta + \Gamma)^2. \quad (33)\end{aligned}$$

We now find a lower bound for the second term in the right hand side of (33). Since the Hessian approximation matrices $\hat{\mathbf{B}}_t$ are positive definite their inverses $\hat{\mathbf{B}}_t^{-1}$ are positive semidefinite. In turn, this implies that all the eigenvalues of $\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}$ are not smaller than Γ since $\Gamma\mathbf{I}$ increases all the eigenvalues of $\hat{\mathbf{B}}_t^{-1}$ by Γ. This lower bound for the eigenvalues of $\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}$ implies that

$$\nabla F(\mathbf{w}_t)^T\left(\hat{\mathbf{B}}_t^{-1} + \Gamma\mathbf{I}\right)\nabla F(\mathbf{w}_t) \geq \Gamma\left\|\nabla F(\mathbf{w}_t)\right\|^2. \quad (34)$$

Substituting the lower bound in (34) for the corresponding summand in (33) and noting the definition of $K := MS^2(1/\delta + \Gamma)^2/2$ in the statement of the lemma, the result in (29) follows.■

Setting aside the term $K\epsilon_t^2$ for the sake of argument (28) defines a supermartingale relationship for the sequence of average functions $F(\mathbf{w}_t)$. This implies that the sequence $\epsilon_t\Gamma\|\nabla F(\mathbf{w}_t)\|^2$ is almost surely summable which, given that the stepsizes ϵ_t are nonsummable as per (5), further implies that the limit infimum $\liminf_{t\to\infty}\|\nabla F(\mathbf{w}_t)\|$ of the gradient norm $\|\nabla F(\mathbf{w}_t)\|$ is almost surely null. This latter observation is equivalent to having $\liminf_{t\to\infty}\|\mathbf{w}_t - \mathbf{w}^*\|^2 = 0$ with probability 1 over realizations of the random samples $\{\tilde{\boldsymbol{\theta}}_t\}_{t=0}^{\infty}$. The term $K\epsilon_t^2$ is a relatively minor nuisance that can be taken care with a technical argument that we present in the proof of the following theorem.

Theorem 1: Consider the RES algorithm as defined by (14)–(17). If Assumptions 1, 2, and 3 hold true and the sequence of stepsizes satisfies (5), the limit infimum of the squared Euclidean distance to optimality $\|\mathbf{w}_t - \mathbf{w}^*\|^2$ satisfies

$$\liminf_{t\to\infty}\|\mathbf{w}_t - \mathbf{w}^*\|^2 = 0 \qquad \text{a.s.} \quad (35)$$

over realizations of the random samples $\{\tilde{\boldsymbol{\theta}}_t\}_{t=0}^{\infty}$.

Proof: The proof uses the relationship in the statement (28) of Lemma 2 to build a supermartingale sequence. For that purpose define the stochastic process γ_t with values

$$\gamma_t := F(\mathbf{w}_t) + K\sum_{u=t}^{\infty}\epsilon_u^2. \quad (36)$$

Note that γ_t is well-defined because $\sum_{u=t}^{\infty}\epsilon_u^2 < \sum_{u=0}^{\infty}\epsilon_u^2 < \infty$ is summable. Further define the sequence β_t with values

$$\beta_t := \epsilon_t\Gamma\left\|\nabla F(\mathbf{w}_t)\right\|^2. \quad (37)$$

Let now $\mathcal{F}_t$ be a sigma-algebra measuring γ_t, β_t, and $\mathbf{w}_t$. The conditional expectation of γ_{t+1} given $\mathcal{F}_t$ can be written as

$$\mathbb{E}\left[\gamma_{t+1}|\mathcal{F}_t\right] = \mathbb{E}\left[F(\mathbf{w}_{t+1})|\mathcal{F}_t\right] + K\sum_{u=t+1}^{\infty}\epsilon_u^2, \quad (38)$$

because the term $K\sum_{u=t}^{\infty}\epsilon_u^2$ is just a deterministic constant. Substituting (28) of Lemma 2 into (38) and using the definitions of γ_t in (36) and β_t in (37) yields

$$\mathbb{E}\left[\gamma_{t+1}|\mathcal{F}_t\right] \leq \gamma_t - \beta_t \quad (39)$$

Since the sequences γ_t and β_t are nonnegative it follows from (39) that they satisfy the conditions of the supermartingale convergence theorem—see e.g. theorem E7.4 [30]. Therefore, we conclude that: (i) The sequence γ_t converges almost surely. (ii) The sum $\sum_{t=0}^{\infty} \beta_t < \infty$ is almost surely finite. Using the explicit form of β_t in (37) we have that $\sum_{t=0}^{\infty} \beta_t < \infty$ is equivalent to

$$\sum_{t=0}^{\infty} \epsilon_t \Gamma \left\| \nabla F(\mathbf{w}_t) \right\|^2 < \infty, \qquad \text{a.s.} \tag{40}$$

Since the sequence of stepsizes is nonsummable for (40) to be true we need to have a vanishing subsequence embedded in $\|\nabla F(\mathbf{w}_t)\|^2$. By definition, this implies that the limit infimum of the sequence $\|\nabla F(\mathbf{w}_t)\|^2$ is null,

$$\liminf_{t\to\infty} \left\| \nabla F(\mathbf{w}_t) \right\|^2 = 0, \qquad \text{a.s.} \tag{41}$$

To transform the gradient bound in (41) into a bound pertaining to the squared distance to optimality $\|\mathbf{w}_t - \mathbf{w}^*\|^2$ simply observe that the lower bound m on the eigenvalues of $\mathbf{H}(\mathbf{w}_t)$ applied to a Taylor's expansion around the optimal argument $\mathbf{w}^*$ implies that

$$F(\mathbf{w}^*) \geq F(\mathbf{w}_t) + \nabla F(\mathbf{w}_t)^T (\mathbf{w}^* - \mathbf{w}_t) + \frac{m}{2} \|\mathbf{w}^* - \mathbf{w}_t\|^2. \tag{42}$$

Observe now that since $\mathbf{w}^*$ is the minimizing argument of $F(\mathbf{w})$ we must have $F(\mathbf{w}^*) - F(\mathbf{w}_t) \leq 0$ for all $\mathbf{w}$. Using this fact and reordering terms we simplify (42) to

$$\frac{m}{2} \|\mathbf{w}^* - \mathbf{w}_t\|^2 \leq \nabla F(\mathbf{w}_t)^T (\mathbf{w}_t - \mathbf{w}^*). \tag{43}$$

Further observe that the Cauchy-Schwarz inequality implies that $\nabla F(\mathbf{w}_t)^T (\mathbf{w}_t - \mathbf{w}^*) \leq \|\nabla F(\mathbf{w}_t)\| \|\mathbf{w}_t - \mathbf{w}^*\|$. Substitution of this bound in (43) and simplification of a $\|\mathbf{w}^* - \mathbf{w}_t\|$ factor yields

$$\frac{m}{2} \|\mathbf{w}_t - \mathbf{w}^*\| \leq \left\| \nabla F(\mathbf{w}_t) \right\|. \tag{44}$$

Since the limit infimum of $\|\nabla F(\mathbf{w}_t)\|$ is null as stated in (41) the result in (35) follows from considering the bound in (44) in the limit as the iteration index $t \to \infty$. ■

Theorem 1 establishes convergence of a subsequence of the RES algorithm summarized in Algorithm 1. In the proof of the prerequisite Lemma 2 the lower bound in the eigenvalues of $\hat{\mathbf{B}}_t$ enforced by the regularization in (17) plays a fundamental role. Roughly speaking, the lower bound in the eigenvalues of $\hat{\mathbf{B}}_t$ results in an upper bound on the eigenvalues of $\hat{\mathbf{B}}_t^{-1}$ which limits the effect of random variations on the stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$. If this regularization is not implemented, i.e., if we keep $\delta = 0$, we may observe catastrophic amplification of random variations of the stochastic gradient. This effect is indeed observed in the numerical experiments in Section IV. The addition of the identity matrix bias $\Gamma\mathbf{I}$ in (14) is instrumental in the proof of Theorem 1 proper. This bias limits the effects of randomness in the curvature estimate $\hat{\mathbf{B}}_t$. If random variations in the curvature estimate $\hat{\mathbf{B}}_t$ result in a matrix $\hat{\mathbf{B}}_t^{-1}$ with small eigenvalues the term $\Gamma\mathbf{I}$ dominates and (14) reduces to (regular) SGD. This ensures continued progress towards the optimal argument $\mathbf{w}^*$.

A. Rate of Convergence

We complement the convergence result in Theorem 1 with a characterization of the expected convergence rate that we introduce in the following theorem.

Theorem 2: Consider the RES algorithm as defined by (14)–(17) and let the sequence of step sizes be given by $\epsilon_t = \epsilon_0 T_0/(T_0 + t)$ with the parameter ϵ_0 sufficiently small and the parameter T_0 sufficiently large so as to satisfy the inequality

$$2\epsilon_0 T_0 \Gamma > 1. \tag{45}$$

If Assumptions 1, 2, and 3 hold true the difference between the expected objective value $\mathbb{E}[F(\mathbf{w}_t)]$ at time t and the optimal objective $F(\mathbf{w}^*)$ satisfies

$$\mathbb{E}\left[F(\mathbf{w}_t)\right] - F(\mathbf{w}^*) \leq \frac{C_0}{T_0 + t}, \tag{46}$$

where the constant C_0 satisfies

$$C_0 = \max\left\{ \frac{\epsilon_0^2 T_0^2 K}{2\epsilon_0 T_0 \Gamma - 1}, \quad T_0 \left(F(\mathbf{w}_0) - F(\mathbf{w}^*)\right) \right\}. \tag{47}$$

Proof: See Appendix. ■

Theorem 2 shows that under specified assumptions, the expected error in terms of the objective value after t RES iterations is at least of order $O(1/t)$. An expected convergence rate of order $O(1/t)$ is typical of stochastic optimization algorithms and, in that sense, no better than conventional SGD. While the convergence rate doesn't change, improvements in convergence time are marked as we illustrate with the numerical experiments of Sections IV and V-A.

IV. Numerical Analysis

We compare convergence times of RES and SGD in problems with small and large condition numbers. We use a stochastic quadratic objective function as a test case. In particular, consider a positive definite diagonal matrix $\mathbf{A} \in \mathbb{S}_n^{++}$, a vector $\mathbf{b} \in \mathbb{R}^n$, a random vector $\boldsymbol{\theta} \in \mathbb{R}^n$, and diagonal matrix $\operatorname{diag}(\boldsymbol{\theta})$ defined by $\boldsymbol{\theta}$. The function $F(\mathbf{w})$ in (1) is defined as

$$\begin{aligned} F(\mathbf{w}) &:= \mathbb{E}_\theta\left[f(\mathbf{w}, \theta)\right] \\ &:= \mathbb{E}_\theta\left[\frac{1}{2}\mathbf{w}^T\left(\mathbf{A} + \mathbf{A}\operatorname{diag}(\boldsymbol{\theta})\right)\mathbf{w} + \mathbf{b}^T\mathbf{w}\right]. \end{aligned} \tag{48}$$

In (48), the vector $\boldsymbol{\theta}$ is chosen uniformly at random from the n-dimensional box $\Theta = [-\theta_0, \theta_0]^n$ for some given constant $\theta_0 < 1$. The linear term $\mathbf{b}^T\mathbf{w}$ is added so that the instantaneous functions $f(\mathbf{w}, \theta)$ have different minima which are (almost surely) different from the minimum of the average function $F(\mathbf{w})$. The quadratic term is chosen so that the condition number of $F(\mathbf{w})$ is the condition number of $\mathbf{A}$. Indeed, just observe that since $\mathbb{E}_\theta[\boldsymbol{\theta}] = \mathbf{0}$, the average function in (48) can be written as $F(\mathbf{w}) = (1/2)\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{b}^T\mathbf{w}$. The parameter θ_0 controls the variability of the instantaneous functions $f(\mathbf{w}, \theta)$. For small $\theta_0 \approx 0$ the instantaneous functions are close to each other and to the average function. For large $\theta_0 \approx 1$ the instantaneous functions vary over a large range. We emphasize that the restriction of $F(\mathbf{w})$ to diagonal positive definite quadratic forms is not significant. What is important is the ability to control the

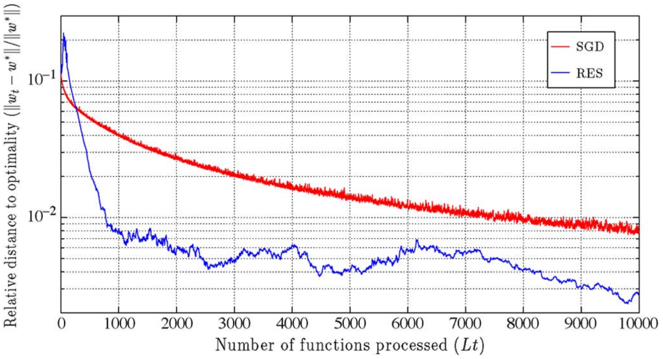


Fig. 1. Convergence of SGD and RES for the function in (48). Relative distance to optimality $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\|$ is shown with respect to the number Lt of stochastic functions processed. For RES the number of iterations required to achieve a certain accuracy is smaller than the corresponding number for SGD. See text for parameters' values.

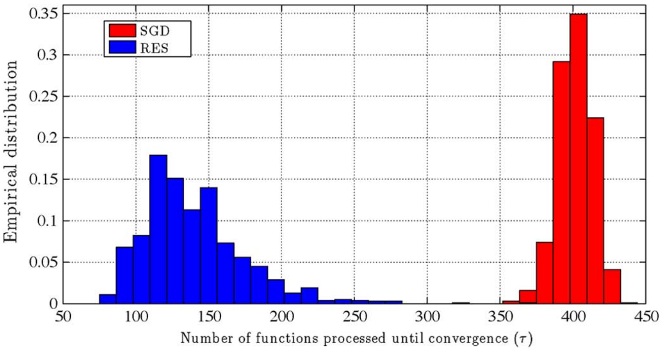


Fig. 2. Convergence of SGD and RES for well-conditioned problems. Empirical distributions of the number $\tau = Lt$ of stochastic functions that are processed to achieve relative precision $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| \leq 10^{-2}$ are shown. Histogram is across $J = 1,000$ realizations of functions as in (48) with condition number $10^{\xi} = 10$. See text for parameters' values.

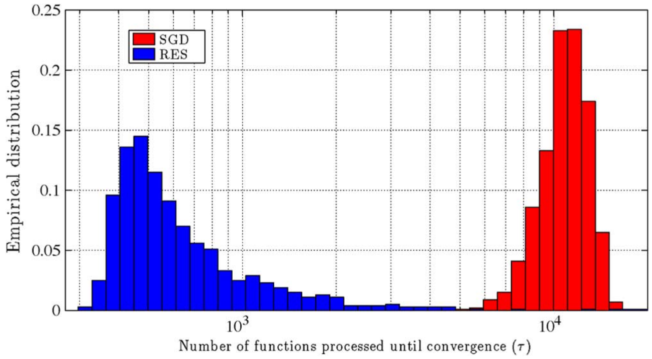


Fig. 3. Convergence of SGD and RES for ill-conditioned problems. Empirical distributions of the number $\tau = Lt$ of stochastic functions that are processed to achieve relative precision $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| \leq 10^{-2}$ are shown. Histogram is across $J = 1,000$ realizations of functions as in (48) with condition number $10^{\xi} = 10^3$. See text for parameters' values.

condition number of $F(\mathbf{w})$ and the variability of the instantaneous functions $f(\mathbf{w}, \theta)$. Analogous results can be obtained if we pre and post multiply the quadratic form with a random orthogonal matrix.

Further note that we can write the optimum argument as $\mathbf{w}^* = \mathbf{A}^{-1}\mathbf{b}$ for comparison against iterates $\mathbf{w}_t$. This allows us to consider a given ρ and study the convergence metric

$$\tau := L \min_t \left\{ t : \frac{\|\mathbf{w}_t - \mathbf{w}^*\|}{\|\mathbf{w}^*\|} \leq \rho \right\}, \tag{49}$$

which represents the time needed to achieve a given relative distance to optimality $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| \leq \rho$ as measured in terms of the number Lt of stochastic functions that are processed to achieve such accuracy.

A. Effect of Problem's Condition Number

To study the effect of the problem's condition number we generate instances of (48) by choosing $\mathbf{b}$ uniformly at random from the box $[0, 1]^n$ and the matrix $\mathbf{A}$ as diagonal with elements a_{ii} uniformly drawn from the discrete set $\{1, 10^{-1}, \ldots, 10^{-\xi}\}$. This choice of $\mathbf{A}$ yields problems with condition number 10^{ξ}.

Representative runs of RES and SGD for $n = 50$, $\theta_0 = 0.5$, and $\xi = 3$ are shown in Fig. 1. For the RES run the stochastic gradients $\hat{\mathbf{s}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ in (3) are computed as an average of $L = 5$ realizations, the regularization parameter in (10) is set to $\delta = 10^{-3}$, and the minimum progress parameter in (14) to $\Gamma = 10^{-4}$. For SGD we use $L = 1$ in (3). In both cases the step size sequence is of the form $\epsilon_t = \epsilon_0 T_0/(T_0 + t)$. Separate rough searches are performed to find step size parameters ϵ_0 and T_0 for RES and SGD that minimize the objective function after 10^4 iterations. For the runs in Fig. 1 the best parameters for SGD are $\epsilon_0 = 10^{-1}$ and $T_0 = 10^3$, while for RES the best choices are $\epsilon_0 = 2 \times 10^{-2}$ and $T_0 = 10^3$. Since we are using different values of L for SGD and RES we plot the relative distance to optimality $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\|$ against the number Lt of functions processed up until iteration t.

As expected for a problem with large condition number—since we are using $\xi = 3$, the condition number of $F(\mathbf{w})$ is 10^3—RES is much faster than SGD. After $t = 10^4$ the distance to optimality for the SGD iterate is $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| = 7.9 \times 10^{-3}$. Comparable accuracy $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| = 7.9 \times 10^{-3}$ for RES is achieved after $t = 179$ iterations. Since we are using $L = 5$ for RES this corresponds to $Lt = 895$ random function evaluations. Conversely, upon processing $Lt = 10^4$ random functions—which corresponds to $t = 2 \times 10^3$ iterations—RES achieves accuracy $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| = 2.7 \times 10^{-3}$. This relative performance difference can be made arbitrarily large by modifying the condition number of $\mathbf{A}$.

A more comprehensive analysis of the relative advantages of RES appears in Figs. 2 and 3. We keep the same parameters used to generate Fig. 1 except that we use $\xi = 1$ for Fig. 2 and $\xi = 3$ for Fig. 3. This yields a family of well-conditioned functions with condition number $10^{\xi} = 10$ and a family of ill-conditioned functions with condition number $10^{\xi} = 10^3$. In Fig. 3 we use the same step size parameters of Fig. 1 because the function's parameters are the same. In Fig. 2, where the condition number is smaller, the best stepsize parameters for SGD are $\epsilon_0 = 6 \times 10^{-1}$ and $T_0 = 10^3$, and for RES the optimal choices are $\epsilon_0 = 10^{-1}$ and $T_0 = 10^3$. In both figures we consider $\rho = 10^{-2}$ and study the convergence times τ and τ' of RES and SGD, respectively [cf. (49)]. Resulting empirical distributions of τ and τ' across $J = 1,000$ instances of the functions $F(\mathbf{w})$ in (48) are reported in Figs. 2 and 3 for the well-conditioned and ill-conditioned families, respectively. For

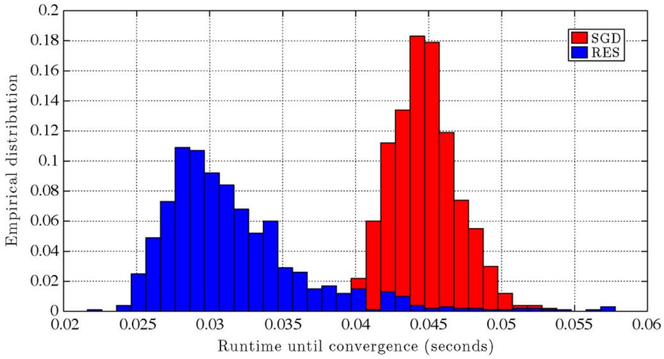


Fig. 4. Central processing unit (CPU) runtimes of SGD and RES for well-conditioned problems. Empirical distributions of CPU runtimes to achieve relative precision $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| \leq 10^{-2}$ are shown. Histogram is across $J = 1,000$ realizations of functions as in (48) with condition number $10^{\xi} = 10^{1}$. See text for parameters' values.

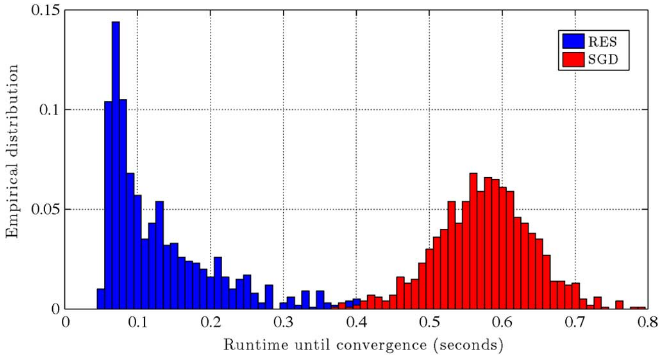


Fig. 5. Central processing unit (CPU) runtimes of SGD and RES for ill-conditioned problems. Empirical distributions of CPU runtimes to achieve relative precision $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| \leq 10^{-2}$ are shown. Histogram is across $J = 1,000$ realizations of functions as in (48) with condition number $10^{\xi} = 10^{3}$. See text for parameters' values.

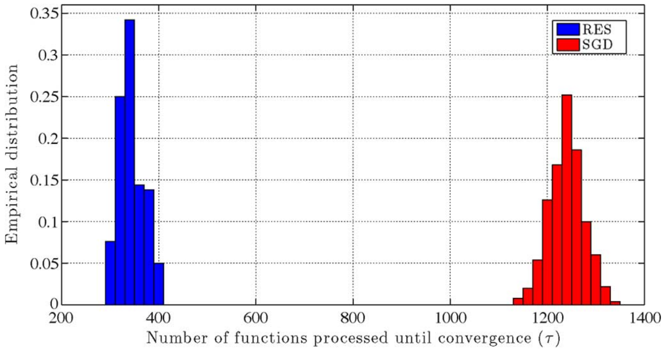


Fig. 6. Convergence of SGD and RES for a very large dimensional problem with small condition number. Empirical distributions of the number Lt of stochastic functions that are processed to achieve relative precision $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| \leq 10^{-2}$ are shown. Histogram is across $J = 1,000$ realizations of functions as in (48) with condition number $10^{\xi} = 10^{1}$. See text for parameters' values.

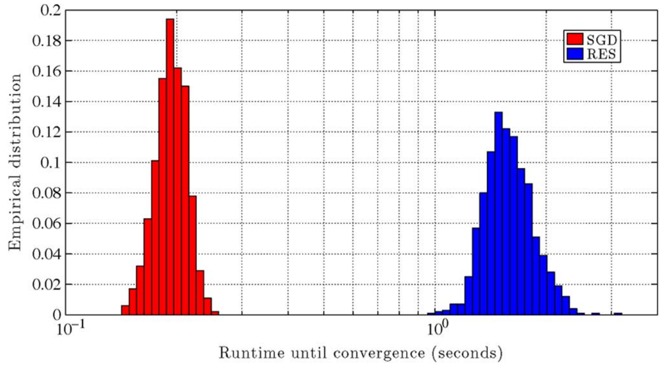


Fig. 7. Convergence of SGD and RES for a very large dimensional problem with small condition number. Empirical distributions of CPU runtimes to achieve relative precision $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| \leq 10^{-2}$ are shown. Histogram is across $J = 1,000$ realizations of functions as in (48) with condition number $10^{\xi} = 10^{1}$. See text for parameters' values.

the well-conditioned family RES reduces the number of functions processed from an average of $\bar{\tau}' = 401$ in the case of SGD to an average of $\bar{\tau} = 139$. This nondramatic improvement becomes more significant for the ill-conditioned family where the reduction is from an average of $\bar{\tau}' = 1.1 \times 10^{4}$ for SGD to an average of $\bar{\tau} = 7.8 \times 10^{2}$ for RES.

B. Central Processing Unit Runtime Comparisons

Since the complexity of each RES iteration is larger than the corresponding complexity of SGD we also compare the performances of SGD and RES in terms of the central processing unit (CPU) runtime required to achieve relative accuracy $\rho = 10^{-2}$. The empirical distributions of runtimes across $J = 1,000$ realizations are reported in Figs. 4 and 5 for the well-conditioned and ill-conditioned families, respectively. In the well-conditioned family, RES reduces runtime from an average of 4.4×10^{-2} seconds in the case of SGD to an average of 3.2×10^{-2} seconds. A more significant improvement can be seen for the ill-conditioned family where the reduction is from an average of 5.7×10^{-1} seconds for SGD to an average of 1.5×10^{-1} seconds for RES.

It is important to emphasize that the advantage of RES in terms of CPU runtime depends on specific problem parameters. In particular, if the condition number of $F(\mathbf{w})$ is small, we expect that as we increase the variable dimension n the RES reduction on the number of iterations is overcome by the added computational complexity of each iteration. To illustrate this drawback we repeat the numerical experiments in Figs. 2 and 4 where the condition number is $10^{\xi} = 10$, but change the number of variables to $n = 5 \times 10^{2}$. Convergence times needed to achieve relative accuracy $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| \leq 10^{-2}$ as measured by the number of stochastic functions processed and the CPU runtime are shown in Figs. 6 and 7, respectively. In both cases we show empirical distributions across $J = 1,000$ realizations of the functions $F(\mathbf{w})$. As evidenced by Fig. 6, RES is faster in terms of the number of random functions required. But, as evidenced by Fig. 7, the opposite is true when we consider CPU runtimes. Indeed, the average number of function evaluations are $\bar{\tau}' = 322$ for RES and $\bar{\tau} = 1.24 \times 10^{3}$ and SGD but the average runtimes are 1.6 seconds for RES and 1.9×10^{-1} seconds for SGD. It follows as a conclusion that SGD outperforms RES for problems that are well-conditioned and large dimensional. We summarize this conclusion in the following remark.

Remark 2: In all of our numerical experiments RES reduces the number of stochastic functions that have to be processed to achieve a target accuracy. The reduction is moderate for well-

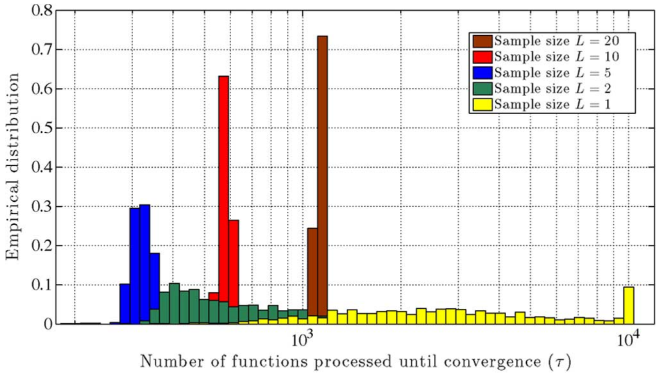


Fig. 8. Convergence of RES for different sample sizes in the computation of stochastic gradients. Empirical distributions of the number $\tau = Lt$ of processed stochastic functions to achieve relative precision $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| \leq 10^{-2}$ are shown when we use $L = 1$, $L = 2$, $L = 5$, $L = 10$, and $L = 20$ in the evaluation of the stochastic gradients $\hat{\mathbf{s}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ in (3). The average convergence time decreases as we go from small to moderate values of L and starts increasing as we go from moderate to large values of L. The variance of convergence times decreases monotonically with increasing L.

conditioned problems but becomes arbitrarily large as the condition number of the objective function increases. However, the computational cost of each RES iteration becomes progressively larger as the dimension of the variable increases. It follows that RES is best suited to problems where the cost of computing stochastic gradients is large, problems where the dimension is not too large, problems where the Hessian approximation matrices are sparse, or problems where the condition number makes SGD impracticable. For problems where the cost of computing stochastic gradients is reasonable, with condition numbers close to one, and whose Hessians lack any amenable structure, SGD and variants of SGD are preferable; see also Section V-A.

C. *Choice of Stochastic Gradient Average*

The stochastic gradients $\hat{\mathbf{s}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ in (3) are computed as an average of L sample gradients $\nabla f(\mathbf{w}, \boldsymbol{\theta}_l)$. To study the effect of the choice of L on RES we consider problems as in (48) with matrices $\mathbf{A}$ and vectors $\mathbf{b}$ generated as in Section IV-A. We consider problems with $n = 50$, $\theta_0 = 0.5$, and $\xi = 2$; set the RES parameters to $\delta = 10^{-3}$ and $\Gamma = 10^{-4}$; and the step size sequence to $\epsilon_t = \epsilon_0 T_0/(T_0 + t)$ with $\epsilon_0 = 10^{-1}$ and $T_0 = 10^3$. We then consider different choices of L and for each specific value generate $J = 1,000$ problem instances. For each run we record the total number τ_L of sample functions that need to be processed to achieve relative distance to optimality $\|\mathbf{w}_t - \mathbf{w}^*\|/\|\mathbf{w}^*\| \leq 10^{-2}$ [cf. (49)]. If $\tau > 10^4$ we report $\tau = 10^4$ and interpret this outcome as a convergence failure. The resulting estimates of the probability distributions of the times τ_L are reported in Fig. 8 for $L = 1$, $L = 2$, $L = 5$, $L = 10$, and $L = 20$.

The trends in convergence times τ apparent in Fig. 8 are: (i) As we increase L the variance of convergence times decreases. (ii) The average convergence time decreases as we go from small to moderate values of L and starts increasing as we go from moderate to large values of L. Indeed, the empirical standard deviations of convergence times decrease monotonically from $\sigma_{\tau_1} = 2.8 \times 10^3$ to $\sigma_{\tau_2} = 2.6 \times 10^2$, $\sigma_{\tau_5} = 31.7$, $\sigma_{\tau_{10}} = 28.8$, and $\sigma_{\tau_{20}} = 22.7$, when L increases from $L = 1$ to $L = 2$, $L = 5$, $L = 10$, and $L = 20$. The empirical mean decreases from $\bar{\tau}_1 = 3.5 \times 10^3$ to $\bar{\tau}_2 = 6.3 \times 10^2$ as we move from $L = 1$ to $L = 2$, stays at about the same value $\bar{\tau}_5 = 3.3 \times 10^2$ for $L = 5$ and then increases to $\bar{\tau}_{10} = 5.8 \times 10^2$ and $\bar{\tau}_{20} = 1.2 \times 10^3$ for $L = 10$ and $L = 20$. This behavior is expected since increasing L results in curvature estimates $\hat{\mathbf{B}}_t$ closer to the Hessian $\mathbf{H}(\mathbf{w}_t)$ thereby yielding better convergence times. As we keep increasing L, there is no payoff in terms of better curvature estimates and we just pay a penalty in terms of more function evaluations for an equally good $\hat{\mathbf{B}}_t$ matrix. This can be corroborated by observing that the convergence times τ_5 are about half those of τ_{10} which in turn are about half those of τ_{20}. This means that the *actual* convergence times τ/L have similar distributions for $L = 5$, $L = 10$, and $L = 20$. The empirical distributions in Fig. 8 show that moderate values of L suffice to provide workable curvature approximations. This justifies the use $L = 5$ in Sections IV-A and IV-D.

D. *Effect of Problem's Dimension*

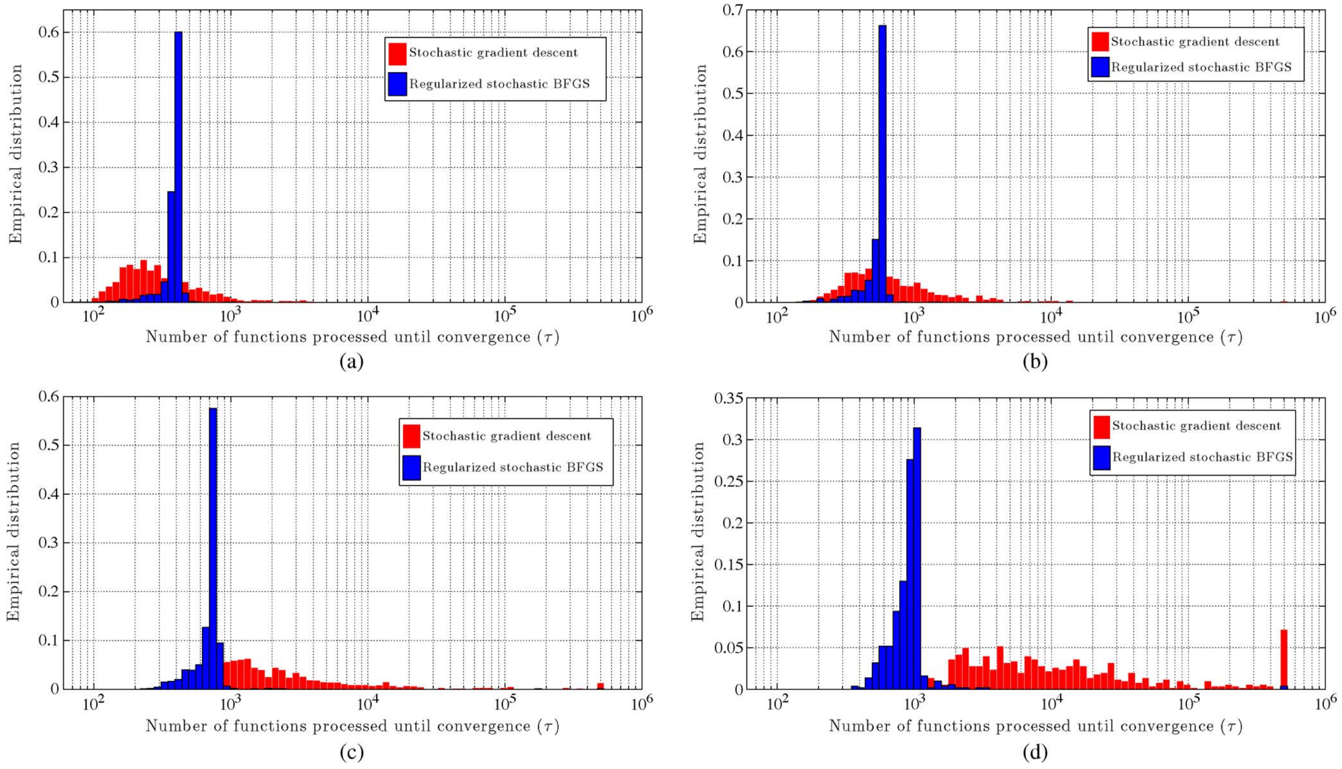
To evaluate performance for problems of different dimensions we consider functions of the form in (48) with $\mathbf{b}$ uniformly chosen from the box $[0, 1]^n$ and diagonal matrix $\mathbf{A}$ as in Section IV-A. However, we select the elements a_{ii} as uniformly drawn from the interval [0,1]. This results in problems with more moderate condition numbers and allows for a comparative study of performance degradations of RES and SGD as the problem dimension n grows.

The variability parameter for the random vector $\boldsymbol{\theta}$ is set to $\theta_0 = 0.5$. The RES parameters are $L = 5$, $\delta = 10^{-3}$, and $\Gamma = 10^{-4}$. For SGD we use $L = 1$. In both methods the step size sequence is $\epsilon_t = \epsilon_0 T_0/(T_0 + t)$ with $\epsilon_0 = 10^{-1}$ and $T_0 = 10^3$. For a problem of dimension n we study convergence times τ_n and τ'_n of RES and SGD as defined in (49) with $\rho = 1$. For each value of n considered we determine empirical distributions of τ_n and τ'_n across $J = 1,000$ problem instances. If $\tau > 5 \times 10^5$ we report $\tau = 5 \times 10^5$ and interpret this outcome as a convergence failure. The resulting histograms are shown in Fig. 9 for $n = 5$, $n = 10$, $n = 20$, and $n = 50$.

For problems of small dimension having $n = 5$ the average performances of RES and SGD are comparable, with SGD performing slightly better. E.g., the medians of these times are $\mathrm{median}(\tau_5) = 400$ and $\mathrm{median}(\tau'_5) = 265$, respectively. A more significant difference is that times τ_5 of RES are more concentrated than times τ'_5 of SGD. The latter exhibits large convergence times $\tau'_5 > 10^3$ with probability 0.06 and fails to converge altogether in a few rare instances—we have $\tau'_5 = 5 \times 10^5$ in 1 out of 1,000 realizations. In the case of RES all realizations of τ_5 are in the interval $70 \leq \tau_5 \leq 1095$.

As we increase n we see that RES retains the smaller spread advantage while eventually exhibiting better average performance as well. Medians for $n = 10$ are still comparable at $\mathrm{median}(\tau_{10}) = 575$ and $\mathrm{median}(\tau'_{10}) = 582$, as well as for $n = 20$ at $\mathrm{median}(\tau_{20}) = 745$ and $\mathrm{median}(\tau'_{20}) = 1427$. For $n = 50$ the RES median is decidedly better since $\mathrm{median}(\tau_{50}) = 950$ and $\mathrm{median}(\tau'_{50}) = 7942$.

For large dimensional problems having $n = 50$ SGD becomes unworkable. It fails to achieve convergence in 5×10^5



(a)

(b)

(c)

(d)

Fig. 9. Histogram of the number of data points that SGD and RES need to converge. Convergence time for RES increases smoothly by increasing the dimension of problem, while convergence time of SGD increases faster. (a) $n = 5$; (b) $n = 10$; (c) $n = 20$; (d) $n = 50$.

iterations with probability 0.07 and exceeds 10^4 iterations {with probability 0.45. For RES we fail to achieve convergence in 5×10^5 iterations with probability 3×10^{-3} and achieve convergence in less than 10^4 iterations in all other cases. Further observe that RES degrades smoothly as n increases. The median number of gradient evaluations needed to achieve convergence increases by a factor of $\text{median}(\tau'_{50})/\text{median}(\tau'_5) = 29.9$ as we increase n by a factor of 10. The spread in convergence times remains stable as n grows.

V. Support Vector Machines

A particular case of (1) is the implementation of a support vector machine (SVM). Given a training set with points whose class is known the goal of an SVM is to find a hyperplane that best separates the training set. To be specific let $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ be a training set containing N pairs of the form $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathbb{R}^n$ is a feature vector and $y_i \in \{-1, 1\}$ is the corresponding vector's class. The goal is to find a hyperplane supported by a vector $\mathbf{w} \in \mathbb{R}^n$ which separates the training set so that $\mathbf{w}^T\mathbf{x}_i > 0$ for all points with $y_i = 1$ and $\mathbf{w}^T\mathbf{x}_i < 0$ for all points with $y_i = -1$. This vector may not exist if the data is not perfectly separable, or, if the data is separable there may be more than one separating vector. We can deal with both situations with the introduction of a loss function $l((\mathbf{x}, y); \mathbf{w})$ defining some measure of distance between the point $\mathbf{x}_i$ and the hyperplane supported by $\mathbf{w}$. We then select the hyperplane supporting vector as

$$\mathbf{w}^* := \underset{\mathbf{w}}{\operatorname{argmin}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} l\left((\mathbf{x}_i, y_i); \mathbf{w}\right), \tag{50}$$

where we also added the regularization term $\lambda\|\mathbf{w}\|^2/2$ for some constant $\lambda > 0$. The vector $\mathbf{w}^*$ in (50) balances the minimization of the sum of distances to the separating hyperplane, as measured by the loss function $l((\mathbf{x}, y); \mathbf{w})$, with the minimization of the L_2 norm $\|\mathbf{w}\|_2$ to enforce desirable properties in $\mathbf{w}^*$. Common selections for the loss function are the hinge loss $l((\mathbf{x}, y); \mathbf{w}) = \max(0, 1 - y(\mathbf{w}^T\mathbf{x}))$, the squared hinge loss $l((\mathbf{x}, y); \mathbf{w}) = \max\left(0, 1 - y(\mathbf{w}^T\mathbf{x})\right)^2$ and the log loss $l((\mathbf{x}, y); \mathbf{w}) = \log(1 + \exp(-y(\mathbf{w}^T\mathbf{x})))$. See, e.g., [4], [26].

In order to model (50) as a stochastic optimization problem in the form of problem (1), we define $\boldsymbol{\theta}_i = (\mathbf{x}_i, y_i)$ as a given training point and $m_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ as a uniform probability distribution on the training set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N} = \{\boldsymbol{\theta}_i\}_{i=1}^{N}$. Upon defining the sample functions

$$f(\mathbf{w}, \boldsymbol{\theta}) = f\left(\mathbf{w}, (\mathbf{x}, y)\right) := \frac{\lambda}{2}\|\mathbf{w}\|^2 + l\left((\mathbf{x}, y); \mathbf{w}\right), \tag{51}$$

it follows that we can rewrite the objective function in (50) as

$$\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} l\left((\mathbf{x}_i, y_i); \mathbf{w}\right) = \mathbb{E}_{\boldsymbol{\theta}}\left[f(\mathbf{w}, \boldsymbol{\theta})\right] \tag{52}$$

since each of the functions $f(\mathbf{w}, \boldsymbol{\theta})$ is drawn with probability $1/N$ according to the definition of $m_{\boldsymbol{\theta}}(\boldsymbol{\theta})$. Substituting (52) into (50) yields a problem with the general form of (1) with random functions $f(\mathbf{w}, \boldsymbol{\theta})$ explicitly given by (51).

We can then use Algorithm (1) to attempt solution of (50). For that purpose we particularize Step 2 to the drawing of L feature vectors $\tilde{\mathbf{x}}_t = [\mathbf{x}_{t1}; \ldots; \mathbf{x}_{tL}]$ and their corresponding class values $\tilde{\mathbf{y}}_t = [y_{t1}; \ldots; y_{tL}]$ to construct the vector of pairs

$\tilde{\boldsymbol{\theta}}_t = [(\mathbf{x}_{t1}, y_{t1}); \ldots; (\mathbf{x}_{tL}, y_{tL})]$. These training points are selected uniformly at random from the training set $\mathcal{S}$. We also need to particularize Steps 3 and 5 to evaluate the stochastic gradient of the instantaneous function in (51). E.g., Step 3 takes the form

$$\begin{aligned}\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t) &= \hat{\mathbf{s}}\left(\mathbf{w}_t, (\tilde{\mathbf{x}}_t, \tilde{\mathbf{y}}_t)\right)\\ &= \lambda \mathbf{w}_t + \frac{1}{L}\sum_{i=1}^{L} \nabla_{\mathbf{w}} l\left((\mathbf{x}_{ti}, y_{ti}); \mathbf{w}_t\right). \qquad (53)\end{aligned}$$

The specific form of Step 5 is obtained by replacing $\mathbf{w}_{t+1}$ for $\mathbf{w}_t$ in (53). We analyze the behavior of Algorithm (1) in the implementation of a SVM in the following section.

A. RES vs Stochastic Gradient Descent for Support Vector Machines

We test Algorithm 1 when using the squared hinge loss $l((\mathbf{x}, y); \mathbf{w}) = \max\left(0, 1 - y(\mathbf{x}^T\mathbf{w})\right)^2$ in (50). The training set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ contains N feature vectors half of which belong to the class $y_i = -1$ with the other half belonging to the class $y_i = 1$. For the class $y_i = -1$ each of the n components of each of the feature vectors $\mathbf{x}_i \in \mathbb{R}^n$ is chosen uniformly at random from the interval [−0.8,0.2]. Likewise, each of the n components of each of the feature vectors $\mathbf{x}_i \in \mathbb{R}^n$ is chosen uniformly at random from the interval [−0.2,0.8] for the class $y_i = 1$. Observe that the overlap in the range of the feature vectors is such that the classification accuracy expected from a clairvoyant classifier that knows the statistical model of the data set is less than 100%.

In all of our numerical experiments the parameter λ in (50) is set to $\lambda = 10^{-4}$. Recall that since the Hessian eigenvalues of $f(\mathbf{w}, \boldsymbol{\theta}) := \lambda\|\mathbf{w}\|^2/2 + l((\mathbf{x}_i, y_i); \mathbf{w})$ are, at least, equal to λ this implies that the eigenvalue lower bound $\tilde{m}$ is such that $\tilde{m} \geq \lambda = 10^{-4}$. We therefore set the RES regularization parameter to $\delta = \lambda = 10^{-4}$. Further set the minimum progress parameter in (3) to $\Gamma = 10^{-4}$.

Accelerated versions of SGD can be used for the implementation of SVMs. We provide a comparison of RES with respect to regular SGD and three accelerated versions: Stochastic Average Gradient (SAG) [13], Semi-Stochastic Gradient Descent (S2GD) [16], and Stochastic Approximation by Averaging (SAA) [14]. The SAG algorithm incorporates memory of previous stochastic gradients and uses an average of stochastic gradients as descent direction. The S2GD algorithm is a hybrid method which runs through several epochs. Each epoch is characterized by the computation of a single full gradient and a random number of stochastic gradients, with the number of stochastic gradients selected according to a geometric distribution. In SAA a time average of iterates is computed and reported.

An illustration of the relative performances of SAA, SGD, SAG, S2GD and RES for $n = 40$ and $N = 10^3$ is presented in Fig. 10. For RES, we set $L = 5$ and choose the decreasing stepsize sequence $\epsilon_t = \epsilon_0 T_0/(T_0 + t)$ with $\epsilon_0 = 4 \times 10^{-1}$ and $T_0 = 10^6$. These parameters yield best performance after processing 10^4 feature vectors. For SGD, SAA, SAG, and S2GD we tune the various parameters and report results for the combination that yields best performance after processing 10^5 feature vectors. In Fig. 10 the value of the objective function $F(\mathbf{w}_t)$ is represented with respect to the number of feature vectors processed, which is given by the product Lt between the iteration index and the sample size used to compute stochastic gradients. To achieve the objective function value $F(\mathbf{w}_t) = 10^{-4}$, RES processes $Lt = 3.3 \times 10^3$ training points which is a little more than 3 passes over the complete data set. The required time for processing these number of feature vectors is 45 milliseconds (ms). Reaching the same objective function value $F(\mathbf{w}_t) = 10^{-4}$ requires processing $Lt = 8.3 \times 10^4$ training points for SGD which is more than 83 passes over the whole data set. It takes 520 ms for SGD to achieve this value for the objective function. The number of processed training points to achieve the same objective function value $F(\mathbf{w}_t) = 10^{-4}$ for SAG and S2GD are $Lt = 4.9 \times 10^4$ and $Lt = 5.2 \times 10^4$, respectively. In terms of CPU runtime SAG and S2GD requires 350 ms and 280 ms to achieve objective function value 10^{-4}. The

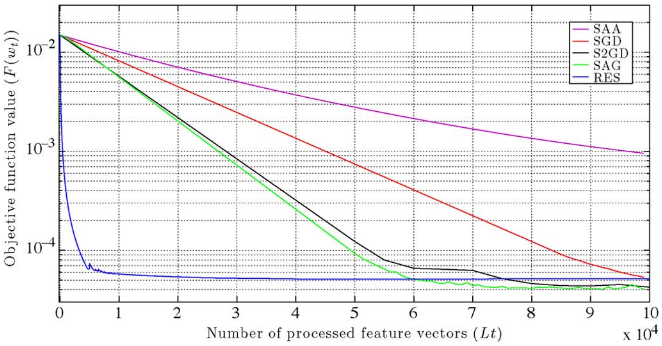


Fig. 10. Comparison of RES, SGD, the SGD accelerations SAA, SAG, and S2GD to find an optimal linear classifier with respect to the cost in (50) for a problem of dimension $n = 40$ and training set with $N = 10^3$ feature vectors. RES processes a much smaller number of feature vectors to achieve comparable objective values. See text for parameters' values.

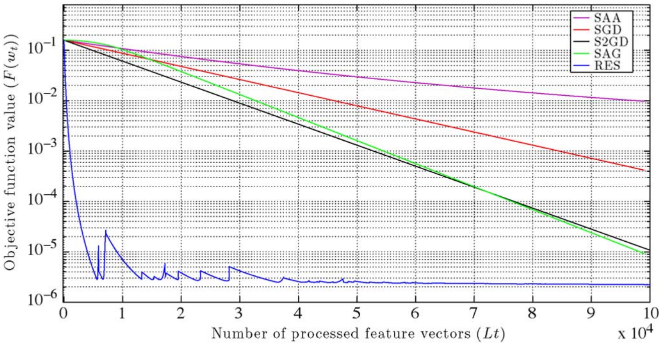


Fig. 11. Comparison of Fig. 10 for problem dimension $n = 400$ and training set cardinality $N = 10^4$. The reduction in the number of feature vectors processed is more pronounced than in Fig. 10, but less pronounced in terms of the CPU runtimes shown in Table I. See text for parameters' values.

TABLE I
CPU RUNTIMES OF RES, SGD, THE SGD ACCELERATIONS SAA, SAG, AND S2GD TO FIND AN OPTIMAL LINEAR CLASSIFIER WITH RESPECT TO THE COST IN (50) FOR DIFFERENT PROBLEM DIMENSION n AND CARDINALITY OF TRAINING SET N. TIMES REPORTED ARE TO ACHIEVE OBJECTIVE VALUES $F(\mathbf{w}_t) = 10^{-4}$

n	N	RES	SGD	SAG	S2GD	SAA
40	10^3	45 ms	520 ms	350 ms	280 ms	> 690 ms
400	10^4	0.8 s	> 1.5 s	1.3 s	1.2 s	> 1.7 s

performance of SAA is worse than the performance of regular SGD.

To compare the performances of RES, SGD, SAA, SAG, and S2GD in a larger SVM problem we set the size of the training set to $N = 10^4$ and the dimension of the feature vectors to $n = 400$. For RES we make $\epsilon_0 = 1 \times 10^{-1}$, $T_0 = 10^1$ and $L = 20$. For SGD and its accelerations we select the parameters that achieve optimal performance after processing 10^4 feature vectors. The results with respect to number of feature vectors processed are shown in Fig. 11 and the CPU times are shown in Table I. The advantage of RES in terms of the number of feature vectors processed is more marked than in the previous experiment. The advantage in terms of CPU processing times is smaller. For reference, RES achieves the objective value $F(\mathbf{w}_t) = 10^{-4}$ after processing 2.1×10^3 feature vectors in 0.8 seconds. Correspondingly, the numbers of feature vectors processed to attain $F(\mathbf{w}_t) = 10^{-4}$ are 7.6×10^4 and 7.7×10^4 for SAG and S2GD. The CPU runtimes are 1.3 and 1.2, respectively. SGD cannot achieve objective function value $F(\mathbf{w}_t) = 10^{-4}$ after processing 10^5 feature vectors in 1.5 seconds. The performance of SAA is still worse than the performance of regular SGD.

B. RES and Stochastic BFGS

We also investigate the difference between regularized and non-regularized versions of stochastic BFGS for feature vectors of dimension $n = 40$. Observe that non-regularized stochastic BFGS corresponds to making $\delta = 0$ and $\Gamma = 0$ in Algorithm 1. To illustrate the advantage of the regularization induced by the proximity requirement in (10), as opposed to the non-regularized proximity requirement in (8), we keep a constant stepsize $\epsilon_t = 10^{-1}$. The corresponding evolutions of the objective function values $F(\mathbf{w}_t)$ with respect to the number of feature vectors processed Lt are shown in Fig. 12 along with the values associated with stochastic gradient descent. As we reach convergence the likelihood of having small eigenvalues appearing in $\hat{\mathbf{B}}_t$ becomes significant. In regularized stochastic BFGS (RES) this results in recurrent jumps away from the optimal classifier $\mathbf{w}^*$. However, the regularization term limits the size of the jumps and further permits the algorithm to consistently recover a reasonable curvature estimate. In Fig. 12 we process 10^4 feature vectors and observe many occurrences of small eigenvalues. However, the algorithm always recovers and heads back to a good approximation of $\mathbf{w}^*$. In the absence of regularization small eigenvalues in $\hat{\mathbf{B}}_t$ result in larger jumps away from $\mathbf{w}^*$. This not only sets back the algorithm by a much larger amount than in the regularized case but also results in a catastrophic deterioration of the curvature approximation matrix $\hat{\mathbf{B}}_t$. In Fig. 12 we observe recovery after the first two occurrences of small eigenvalues but eventually there is a catastrophic deviation after which non-regularized stochastic BFGS behaves not better than SGD.

VI. Conclusions

Strongly convex optimization problems with stochastic objectives were considered. RES, a stochastic implementation of a regularized version of the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton method was introduced to find corresponding optimal arguments. Almost sure convergence of at least a subsequence generated by RES was established under the assumption that sample functions have well-behaved Hessians. A linear convergence rate in expectation was further proven. Numerical results showed that RES affords important reductions in terms of convergence time relative to stochastic gradient descent. These reductions are of particular significance for problems with large condition numbers or large dimensionality since RES exhibits remarkable stability in terms of the total number of iterations required to achieve target accuracies. An application of RES to support vector machines was also developed. In this particular case the advantages of RES manifest in improvements of classification accuracies for training sets of fixed cardinality. Future research directions include the development of limited memory versions as well as distributed versions where the function to be minimized is spread over agents of a network.

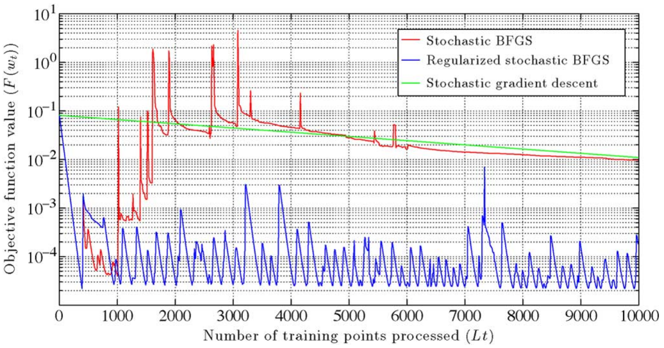


Fig. 12. Comparison of SGD, regularized stochastic BFGS (RES), and (non regularized) stochastic BFGS. The regularization is fundamental to control the erratic behavior of stochastic BFGS; See text for parameters' values.

Appendix A
Proof of Proposition 1

We first show that (13) is true. Since the optimization problem in (10) is convex in $\mathbf{Z}$ we can determine the optimal variable $\mathbf{B}_{t+1} = \mathbf{Z}^*$ using Lagrangian duality. Introduce then the multiplier variable $\boldsymbol{\mu}$ associated with the secant constraint $\mathbf{Z}\mathbf{v}_t = \mathbf{r}_t$ in (10) and define the Lagrangian

$$\mathcal{L}(\mathbf{Z}, \boldsymbol{\mu}) = \operatorname{tr}\left(\mathbf{B}_t^{-1}(\mathbf{Z} - \delta\mathbf{I})\right) - \log\det\left(\mathbf{B}_t^{-1}(\mathbf{Z} - \delta\mathbf{I})\right) - n + \boldsymbol{\mu}^T\left(\mathbf{Z}\mathbf{v}_t - \mathbf{r}_t\right). \quad (54)$$

The dual function is defined as $d(\boldsymbol{\mu}) := \min_{\mathbf{Z}\succeq\mathbf{0}} \mathcal{L}(\mathbf{Z}, \boldsymbol{\mu})$ and the optimal dual variable is $\boldsymbol{\mu}^* := \operatorname{argmin}_{\boldsymbol{\mu}} d(\boldsymbol{\mu})$. We define the primal Lagrangian minimizer associated with dual variable $\boldsymbol{\mu}$ as

$$\mathbf{Z}(\boldsymbol{\mu}) := \operatorname*{argmin}_{\mathbf{Z}\succeq\mathbf{0}} \mathcal{L}(\mathbf{Z}, \boldsymbol{\mu}). \quad (55)$$

Observe that combining the definitions in (55) and (54) we can write the dual function $d(\boldsymbol{\mu})$ as

$$\begin{aligned} d(\boldsymbol{\mu}) &= \mathcal{L}\left(\mathbf{Z}(\boldsymbol{\mu}), \boldsymbol{\mu}\right) \\ &= \operatorname{tr}\left(\mathbf{B}_t^{-1}\left(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I}\right)\right) - \log\det\left(\mathbf{B}_t^{-1}\left(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I}\right)\right) \\ &\quad - n + \boldsymbol{\mu}^T\left(\mathbf{Z}(\boldsymbol{\mu})\mathbf{v}_t - \mathbf{r}_t\right). \quad (56) \end{aligned}$$

We will determine the optimal Hessian approximation $\mathbf{Z}^* = \mathbf{Z}(\boldsymbol{\mu}^*)$ as the Lagrangian minimizer associated with the optimal dual variable $\boldsymbol{\mu}^*$. To do so we first find the Lagrangian minimizer

(55) by nulling the gradient of $\mathcal{L}(\mathbf{Z},\boldsymbol{\mu})$ with respect to $\mathbf{Z}$ in order to show that $\mathbf{Z}(\boldsymbol{\mu})$ must satisfy

$$\mathbf{B}_t^{-1} - \left(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I}\right)^{-1} + \frac{\boldsymbol{\mu}\mathbf{v}_t^T + \mathbf{v}_t\boldsymbol{\mu}^T}{2} = 0. \tag{57}$$

Multiplying the equality in (57) by $\mathbf{B}_t$ from the right and rearranging terms it follows that the inverse of the argument of the log-determinant function in (56) can be written as

$$\left(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I}\right)^{-1}\mathbf{B}_t = \mathbf{I} + \left(\frac{\boldsymbol{\mu}\mathbf{v}_t^T + \mathbf{v}_t\boldsymbol{\mu}^T}{2}\right)\mathbf{B}_t. \tag{58}$$

If, instead, we multiply (57) by $(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I})$ from the right it follows after rearranging terms that

$$\mathbf{B}_t^{-1}\left(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I}\right) = \mathbf{I} - \frac{\boldsymbol{\mu}\mathbf{v}_t^T + \mathbf{v}_t\boldsymbol{\mu}^T}{2}\left(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I}\right). \tag{59}$$

Further considering the trace of both sides of (59) and noting that $\operatorname{tr}(\mathbf{I}) = n$ we can write the trace in (56) as

$$\operatorname{tr}\left(\mathbf{B}_t^{-1}\left(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I}\right)\right) = n - \operatorname{tr}\left[\frac{\boldsymbol{\mu}\mathbf{v}_t^T + \mathbf{v}_t\boldsymbol{\mu}^T}{2}\left(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I}\right)\right]. \tag{60}$$

Observe now that since the trace of a product is invariant under cyclic permutations of its arguments and the matrix $\mathbf{Z}$ is symmetric we have $\operatorname{tr}[\boldsymbol{\mu}\mathbf{v}_t^T(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I})]$ $=\operatorname{tr}[\mathbf{v}\boldsymbol{\mu}_t^T(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I})] = \operatorname{tr}[\boldsymbol{\mu}^T(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I})\mathbf{v}_t]$. Since the argument in the latter is a scalar the trace operation is inconsequential from which it follows that we can rewrite (60) as

$$\operatorname{tr}\left(\mathbf{B}_t^{-1}\left(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I}\right)\right) = n - \boldsymbol{\mu}^T\left(\mathbf{Z}(\boldsymbol{\mu}) - \delta\mathbf{I}\right)\mathbf{v}_t. \tag{61}$$

Observing that the log-determinant of a matrix is the opposite of the log-determinant of its inverse we can substitute (58) for the argument of the log-determinant in (56). Further substituting (61) for the trace in (56) and rearranging terms yields the explicit expression for the dual function

$$d(\boldsymbol{\mu}) = \log\det\left[\mathbf{I} + \left(\frac{\boldsymbol{\mu}\mathbf{v}_t^T + \mathbf{v}_t\boldsymbol{\mu}^T}{2}\right)\mathbf{B}_t\right] - \boldsymbol{\mu}^T(\mathbf{r}_t - \delta\mathbf{v}_t). \tag{62}$$

In order to compute the optimal dual variable $\boldsymbol{\mu}^*$ we set the gradient of (62) to zero and manipulate terms to obtain

$$\boldsymbol{\mu}^* = \frac{1}{\tilde{\mathbf{r}}_t^T\mathbf{v}_t}\left(\mathbf{v}_t\left(1 + \frac{\tilde{\mathbf{r}}_t^T\mathbf{B}_t^{-1}\tilde{\mathbf{r}}_t}{\tilde{\mathbf{r}}_t^T\mathbf{v}_t}\right) - 2\mathbf{B}_t^{-1}\tilde{\mathbf{r}}_t\right), \tag{63}$$

where we have used the definition of the corrected gradient variation $\tilde{\mathbf{r}}_t := \mathbf{r}_t - \delta\mathbf{v}_t$. To complete the derivation plug the expression for the optimal multiplier $\boldsymbol{\mu}^*$ in (63) into the Lagrangian minimizer expression in (57) and regroup terms so as to write

$$\left(\mathbf{Z}(\boldsymbol{\mu}^*) - \delta\mathbf{I}\right)^{-1} = \frac{\mathbf{v}_t\mathbf{v}_t^T}{\tilde{\mathbf{r}}_t^T\mathbf{v}_t} + \left(\mathbf{I} - \frac{\mathbf{v}_t\tilde{\mathbf{r}}_t^T}{\tilde{\mathbf{r}}_t^T\mathbf{v}_t}\right)\mathbf{B}_t^{-1}\left(\mathbf{I} - \frac{\tilde{\mathbf{r}}_t\mathbf{v}_t^T}{\tilde{\mathbf{r}}_t^T\mathbf{v}_t}\right). \tag{64}$$

Applying the Sherman-Morrison formula to compute the inverse of the right hand side of (64) leads to

$$\mathbf{Z}(\boldsymbol{\mu}^*) - \delta\mathbf{I} = \mathbf{B}_t + \frac{\tilde{\mathbf{r}}_t\tilde{\mathbf{r}}_t^T}{\mathbf{v}_t^T\tilde{\mathbf{r}}_t} - \frac{\mathbf{B}_t\mathbf{v}_t\mathbf{v}_t^T\mathbf{B}_t}{\mathbf{v}_t^T\mathbf{B}_t\mathbf{v}_t}, \tag{65}$$

which can be verified by direct multiplication. The result in (13) follows after solving (65) for $\mathbf{Z}(\boldsymbol{\mu}^*)$ and noting that for the convex optimization problem in (10) we must have $\mathbf{Z}(\boldsymbol{\mu}^*) = \mathbf{Z}^* = \mathbf{B}_{t+1}$ as we already argued.

To prove (12) we operate directly from (13). Consider first the term $\tilde{\mathbf{r}}_t\tilde{\mathbf{r}}_t^T/\mathbf{v}_t^T\tilde{\mathbf{r}}_t$ and observe that since the hypotheses include the condition $\mathbf{v}_t^T\tilde{\mathbf{r}}_t > 0$, we must have

$$\frac{\tilde{\mathbf{r}}_t\tilde{\mathbf{r}}_t^T}{\mathbf{v}_t^T\tilde{\mathbf{r}}_t} \succeq \mathbf{0}. \tag{66}$$

Consider now the term $\mathbf{B}_t - \mathbf{B}_t\mathbf{v}_t\mathbf{v}_t^T\mathbf{B}_t/\mathbf{v}_t^T\mathbf{B}_t\mathbf{v}_t$ and factorize $\mathbf{B}_t^{1/2}$ from the left and right side so as to write

$$\mathbf{B}_t - \frac{\mathbf{B}_t\mathbf{v}_t\mathbf{v}_t^T\mathbf{B}_t}{\mathbf{v}_t^T\mathbf{B}_t\mathbf{v}_t} = \mathbf{B}_t^{\frac{1}{2}}\left(\mathbf{I} - \frac{\mathbf{B}_t^{\frac{1}{2}}\mathbf{v}_t\mathbf{v}_t^T\mathbf{B}_t^{\frac{1}{2}}}{\mathbf{v}_t^T\mathbf{B}_t\mathbf{v}_t}\right)\mathbf{B}_t^{\frac{1}{2}} \tag{67}$$

Define the vector $\mathbf{u}_t := \mathbf{B}_t^{1/2}\mathbf{v}_t$ and write $\mathbf{v}_t^T\mathbf{B}_t\mathbf{v}_t = (\mathbf{B}_t^{1/2}\mathbf{v}_t)^T(\mathbf{B}_t^{1/2}\mathbf{v}_t) = \mathbf{u}_t^T\mathbf{u}_t$ as well as $\mathbf{B}_t^{1/2}\mathbf{v}_t\mathbf{v}_t^T\mathbf{B}_t^{1/2} = \mathbf{u}_t\mathbf{u}_t^T$. Substituting these observations into (67) we can conclude that

$$\mathbf{B}_t - \frac{\mathbf{B}_t\mathbf{v}_t\mathbf{v}_t^T\mathbf{B}_t}{\mathbf{v}_t^T\mathbf{B}_t\mathbf{v}_t} = \mathbf{B}_t^{\frac{1}{2}}\left(\mathbf{I} - \frac{\mathbf{u}_t\mathbf{u}_t^T}{\mathbf{u}_t^T\mathbf{u}_t}\right)\mathbf{B}_t^{\frac{1}{2}} \succeq \mathbf{0}, \tag{68}$$

because the eigenvalues of the matrix $\mathbf{u}_t\mathbf{u}_t^T/\mathbf{u}_t^T\mathbf{u}_t$ belong to the interval [0,1]. The only term in (13) which has not been considered is $\delta\mathbf{I}$. Since the rest add up to a positive semidefinite matrix it then must be that (12) is true.

APPENDIX B
PROOF OF THEOREM 2

Theorem 2 claims that the sequence of expected objective values $\mathbb{E}[F(\mathbf{w}_t)]$ approaches the optimal objective $F(\mathbf{w}^*)$ at a linear rate $O(1/t)$. Before proceeding to the proof of Theorem 2 we introduce a technical lemma that provides a sufficient condition for a sequence u_t to exhibit a linear convergence rate.

Lemma 3: Let $c > 1$, $b > 0$ and $t_0 > 0$ be given constants and $u_t \geq 0$ be a nonnegative sequence that satisfies the inequality

$$u_{t+1} \leq \left(1 - \frac{c}{t+t_0}\right)u_t + \frac{b}{(t+t_0)^2}, \tag{69}$$

for all times $t \geq 0$. The sequence u_t is then bounded as

$$u_t \leq \frac{Q}{t+t_0}, \tag{70}$$

for all times $t \geq 0$, where the constant Q is defined as

$$Q := \max\left[\frac{b}{c-1},\, t_0u_0\right]. \tag{71}$$

Proof: We prove (70) using induction. To prove the claim for $t = 0$ simply observe that the definition of Q in (71) implies that

$$Q := \max\left[\frac{b}{c-1},\, t_0u_0\right] \geq t_0u_0, \tag{72}$$

because the maximum of two numbers is at least equal to both of them. By rearranging the terms in (72) we can conclude that

$$u_0 \leq \frac{Q}{t_0}. \tag{73}$$

Comparing (73) and (70) it follows that the latter inequality is true for $t = 0$.

Introduce now the induction hypothesis that (70) is true for $t = s$. To show that this implies that (70) is also true for $t = s+1$ substitute the induction hypothesis $u_s \leq Q/(s + t_0)$ into the recursive relationship in (69). This substitution shows that u_{s+1} is bounded as

$$u_{s+1} \leq \left(1 - \frac{c}{s+t_0}\right)\frac{Q}{s+t_0} + \frac{b}{(s+t_0)^2}. \tag{74}$$

Observe now that according to the definition of Q in (71), we know that $b/(c-1) \leq Q$ because Q is the maximum of $b/(c-1)$ and $t_0 u_0$. Reorder this bound to show that $b \leq Q(c-1)$ and substitute into (74) to write

$$u_{s+1} \leq \left(1 - \frac{c}{s+t_0}\right)\frac{Q}{s+t_0} + \frac{(c-1)Q}{(s+t_0)^2}. \tag{75}$$

Pulling out $Q/(s + t_0)^2$ as a common factor and simplifying and reordering terms it follows that (75) is equivalent to

$$u_{s+1} \leq \frac{Q\left[s + t_0 - c + (c-1)\right]}{(s+t_0)^2} = \frac{s+t_0-1}{(s+t_0)^2}Q. \tag{76}$$

To complete the induction step use the difference of squares formula for $(s + t_0)^2 - 1$ to conclude that

$$[(s+t_0) - 1][(s+t_0)+1] = (s+t_0)^2 - 1 \leq (s+t_0)^2. \tag{77}$$

Reordering terms in (77) it follows that $[(s+t_0)-1]/(s+t_0)^2 \leq 1/[(s+t_0)+1]$, which upon substitution into (76) leads to the conclusion that

$$u_{s+1} \leq \frac{Q}{s+t_0+1}. \tag{78}$$

Eq. (78) implies that the assumed validity of (70) for $t = s$ implies the validity of (70) for $t = s + 1$. Combined with the validity of (70) for $t = 0$, which was already proved, it follows that (70) is true for all times $t \geq 0$. ■

Lemma 3 shows that satisfying (69) is sufficient for a sequence to have the linear rate of convergence specified in (70). In the following proof of Theorem 2 we show that if the step-size sequence parameters ϵ_0 and T_0 satisfy (45) the sequence $\mathbb{E}[F(\mathbf{w}_t)] - F(\mathbf{w}^*)$ of expected optimality gaps satisfies (69) with $c = 2\epsilon_0 T_0 \Gamma$, $b = \epsilon_0^2 T_0^2 K$ and $t_0 = T_0$. The result in (46) then follows as a direct consequence of Lemma 3.

Proof of Theorem 2: Consider the result in (28) of Lemma 2 and subtract the average function optimal value $F(\mathbf{w}^*)$ from both sides of the inequality to conclude that the sequence of optimality gaps in the RES algorithm satisfies

$$\begin{aligned}&\mathbb{E}\left[F(\mathbf{w}_{t+1})|\mathbf{w}_t\right] - F(\mathbf{w}^*)\\ &\qquad \leq F(\mathbf{w}_t) - F(\mathbf{w}^*) - \epsilon_t\Gamma\left\|\nabla F(\mathbf{w}_t)\right\|^2 + \epsilon_t^2 K, \end{aligned}\tag{79}$$

where, we recall, $K := MS^2((1/\delta) + \Gamma)^2/2$ by definition.

We proceed to find a lower bound for the gradient norm $\|\nabla F(\mathbf{w}_t)\|$ in terms of the error of the objective value $F(\mathbf{w}_t) - F(\mathbf{w}^*)$—this is a standard derivation which we include for completeness, see, e.g., [31]. It follows from Assumption 1 that the eigenvalues of the Hessian $\mathbf{H}(\mathbf{w}_t)$ are bounded between $0 < m$ and $M < \infty$ as stated in (22). Taking a Taylor's expansion of the objective function $F(\mathbf{y})$ around $\mathbf{w}$ and using the lower bound in the Hessian eigenvalues we can write

$$F(\mathbf{y}) \geq F(\mathbf{w}) + \nabla F(\mathbf{w})^T(\mathbf{y} - \mathbf{w}) + \frac{m}{2}\|\mathbf{y} - \mathbf{w}\|^2. \tag{80}$$

For fixed $\mathbf{w}$, the right hand side of (80) is a quadratic function of $\mathbf{y}$ whose minimum argument we can find by setting its gradient to zero. Doing this yields the minimizing argument $\hat{\mathbf{y}} = \mathbf{w} - (1/m)\nabla F(\mathbf{w})$ implying that for all $\mathbf{y}$ we must have

$$\begin{aligned}F(\mathbf{y}) &\geq F(\mathbf{w}) + \nabla F(\mathbf{w})^T(\hat{\mathbf{y}} - \mathbf{w}) + \frac{m}{2}\|\hat{\mathbf{y}} - \mathbf{w}\|^2\\ &= F(\mathbf{w}) - \frac{1}{2m}\left\|\nabla F(\mathbf{w})\right\|^2. \end{aligned}\tag{81}$$

The bound in (81) is true for all $\mathbf{w}$ and $\mathbf{y}$. In particular, for $\mathbf{y} = \mathbf{w}^*$ and $\mathbf{w} = \mathbf{w}_t$ (81) yields

$$F(\mathbf{w}^*) \geq F(\mathbf{w}_t) - \frac{1}{2m}\left\|\nabla F(\mathbf{w}_t)\right\|^2. \tag{82}$$

Rearrange terms in (82) to obtain a bound on the gradient norm squared $\|\nabla F(\mathbf{w}_t)\|^2$. Further substitute the result in (79) and regroup terms to obtain the bound

$$\begin{aligned}&\mathbb{E}\left[F(\mathbf{w}_{t+1})|\mathbf{w}_t\right] - F(\mathbf{w}^*)\\ &\qquad \leq \left(1 - 2m\epsilon_t\Gamma\right)\left(F(\mathbf{w}_t) - F(\mathbf{w}^*)\right) + \epsilon_t^2 K. \end{aligned}\tag{83}$$

Take now expected values on both sides of (83). The resulting double expectation in the left hand side simplifies to $\mathbb{E}[\mathbb{E}[F(\mathbf{w}_{t+1})|\mathbf{w}_t]] = \mathbb{E}[F(\mathbf{w}_{t+1})]$, which allow us to conclude that (83) implies that

$$\begin{aligned}&\mathbb{E}\left[F(\mathbf{w}_{t+1})\right] - F(\mathbf{w}^*)\\ &\qquad \leq \left(1 - 2m\epsilon_t\Gamma\right)\left(\mathbb{E}\left[F(\mathbf{w}_t)\right] - F(\mathbf{w}^*)\right) + \epsilon_t^2 K. \end{aligned}\tag{84}$$

Further substituting $\epsilon_t = \epsilon_0 T_0/(T_0 + t)$, which is the assumed form of the step size sequence by hypothesis, we can rewrite (84) as

$$\begin{aligned}&\mathbb{E}\left[F(\mathbf{w}_{t+1})\right] - F(\mathbf{w}^*)\\ &\leq\left(1 - \frac{2\epsilon_0 T_0 \Gamma}{(T_0 + t)}\right)\left(\mathbb{E}\left[F(\mathbf{w}_t)\right] - F(\mathbf{w}^*)\right) + \left(\frac{\epsilon_0 T_0}{T_0 + t}\right)^2 K. \end{aligned}\tag{85}$$

Given that the product $2\epsilon_0 T_0 \Gamma > 1$ as per the hypothesis in (45) the sequence $\mathbb{E}[F(\mathbf{w}_{t+1})] - F(\mathbf{w}^*)$ satisfies the hypotheses of Lemma 3 with $c = 2\epsilon_0 T_0 \Gamma$, $b = \epsilon_0^2 T_0^2 K$ and $t_0 = T_0$. It then follows from (70) and (71) that (46) is true for the constant C_0 defined in (47) upon identifying u_t with $\mathbb{E}[F(\mathbf{x}_{t+1})] - F(\mathbf{x}^*)$, C_0 with Q, and substituting $c = 2\epsilon_0 T_0 \Gamma$, $b = \epsilon_0^2 T_0^2 K$ and $t_0 = T_0$ for their explicit values.

References

[1] A. Mokhtari and A. Ribeiro, "Regularized Stochastic BFGS algorithm," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Austin, TX, USA, Dec. 3–5, 2013, pp. 1109–1112.

[2] A. Mokhtari and A. Ribeiro, "A quasi-newton method for large scale support vector machines," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Florence, Italy, May 4–9, 2014, pp. 8302–8306.

[3] L. Bottou and Y. L. Cun, "On-line learning for very large datasets," *Appl. Stoch. Models Bus. Ind.*, vol. 21, pp. 137–151, 2005.

[4] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, 2010, pp. 177–186, Physica-Verlag HD.

[5] S. Shalev-Shwartz and N. Srebro, "SVM optimization: Inverse dependence on training set size," in *Proc. 25th ACM Int. Conf. Mach. Learn.*, 2008, pp. 928–935.

[6] A. Mokhtari and A. Ribeiro, "A dual stochastic DFP algorithm for optimal resource allocation in wireless systems," in *Proc. IEEE 14th Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Darmstadt, Germany, Jun. 16–19, 2013, pp. 21–25.

[7] A. Ribeiro, "Ergodic stochastic optimization algorithms for wireless communication and networking," *IEEE Trans. Signal Process.*, vol. 58, no. 12, pp. 6369–6386, Dec. 2010.

[8] A. Ribeiro, "Optimal resource allocation in wireless communication and networking," *EURASIP J. Wireless Commun.*, vol. 2012, no. 272, pp. 3727–3741, Aug. 2012.

[9] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *Proc. 24th ACM Int. Conf. Mach. Learn.*, 2007, pp. 807–814.

[10] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proc. ACM 21st Int. Conf. Mach. Learn.*, 2004, pp. 919–926.

[11] A. Nemirovski, A. Juditsky, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1574–1609, 2009.

[12] N. L. Roux, M. Schmidt, and F. R. Bach, "A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets," *Adv. Neural Inf. Process. Syst.*, pp. 2663–2671, 2012.

[13] M. Schmidt, N. L. Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," 2013, arXiv preprint 1309.2388 [Online]. Available: http://arxiv.org/pdf/1309.2388.pdf

[14] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM J. Control Optim.*, vol. 30, no. 4, pp. 838–855, 1992.

[15] A. R. W. Syski, "Stochastic approximation method with gradient averaging for unconstrained problems," *IEEE Trans. Autom. Control*, vol. 28, no. 12, pp. 1097–1105, 1983.

[16] J. Konecny and P. Richtarik, "Semi-stochastic gradient descent methods," 2013, arXiv preprint 1312.1666 [Online]. Available: http://arxiv.org/pdf/1312.1666.pdf

[17] L. Zhang, M. Mahdavi, and R. Jin, "Linear convergence with condition number independent access of full gradients," *Adv. Neural Inf. Process. Syst.*, pp. 980–988, 2013.

[18] J. R. Birge, X. Chen, L. Qi, and Z. Wei, "A stochastic Newton method for stochastic quadratic programs with resource," Univ. of Michigan, Ann Arbor, MI, USA, Tech. Rep., 1995.

[19] M. Zargham, A. Ribeiro, and A. Jadbabaie, "Accelerated backpressure algorithm," 2013, arXiv preprint 1302.1475 [Online]. Available: http://arxiv.org/pdf/1302.1475v1.pdf

[20] J. J. E. Dennis and J. J. More, "A characterization of super linear convergence and its application to quasi-newton methods," *Math. Comput.*, vol. 28, no. 126, pp. 549–560, 1974.

[21] M. J. D. Powell, *Some Global Convergence Properties of a Variable Metric Algorithm for Minimization Without Exact Line Search*, 2nd ed. London, U.K.: Academic, 1971.

[22] R. H. Byrd, J. Nocedal, and Y. Yuan, "Global convergence of a class of quasi-newton methods on convex problems," *SIAM J. Numer. Anal.*, vol. 24, no. 5, pp. 1171–1190, Oct. 1987.

[23] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer-Verlag, 1999.

[24] N. N. Schraudolph, J. Yu, and S. Gunter, "A stochastic quasi-newton method for online convex optimization," in *Proc. 11th Int. Conf. Artif. Intell. Statist. (AIstats)*, 2007, pp. 433–440, Soc. for Artificial Intelligence and Statistics.

[25] A. Bordes, L. Bottou, and P. Gallinari, "SGD-QN: Careful quasi-Newton stochastic gradient descent," *J. Mach. Learn. Res.*, vol. 10, pp. 1737–1754, 2009.

[26] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. New York, NY, USA: Springer, 1999.

[27] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 144–152, ACM.

[28] C. G. Broyden, J. E. D. Wang, Jr., and J. J. More, "On the local and superlinear convergence of quasi-Newton methods," *IMA J. Appl. Math.*, vol. 12, no. 3, pp. 223–245, Jun. 1973.

[29] R. Fletcher, *Practical Methods of Optimizations*. New York, NY, USA: Wiley, 2013.

[30] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.

[31] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Aryan Mokhtari received the B. Eng. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2011, and the M.S. degree in electrical engineering from University of Pennsylvania, Philadelphia, PA, in 2014. Since 2012, he has been working towards the Ph.D. degree in the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA. His research interests include stochastic optimization, machine learning and distributed optimization.

Alejandro Ribeiro (S'02–M'07) received the B.Sc. degree in electrical engineering from the Universidad de la Republica Oriental del Uruguay, Montevideo, in 1998 and the M.Sc. and Ph.D. degrees in electrical engineering from the Department of Electrical and Computer Engineering, the University of Minnesota, Minneapolis, in 2005 and 2007, respectively. From 1998 to 2003, he was a member of the technical staff at Bellsouth Montevideo. In 2008, he joined the University of Pennsylvania (Penn), Philadelphia, where he is currently an Associate Professor at the Department of Electrical and Systems Engineering. His research interests are in the applications of statistical signal processing to the study of networks and networked phenomena. His current research focuses on wireless networks, network optimization, learning in networks, networked control, robot teams, and structured representations of networked data structures. Dr. Ribeiro received the 2012 S. Reid Warren, Jr. Award presented by Penn's undergraduate student body for outstanding teaching and the NSF CAREER Award in 2010. He is also a Fulbright scholar and the recipient of student paper awards at the 2013 American Control Conference (as adviser), as well as the 2005 and 2006 International Conferences on Acoustics, Speech, and Signal Processing.