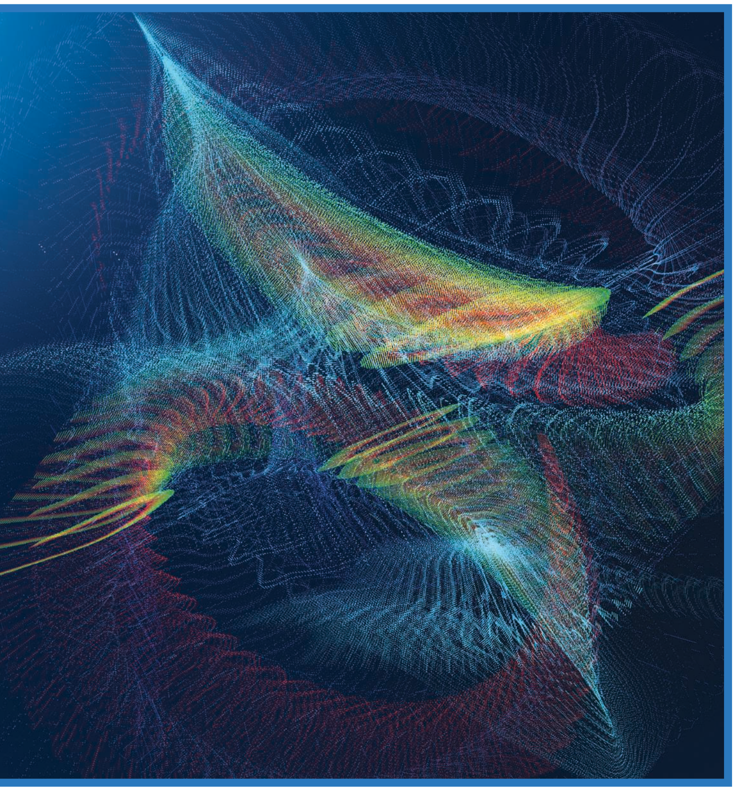
Frank E. Curtis and Katya Scheinberg

Adaptive Stochastic Optimization

A framework for analyzing stochastic optimization algorithms



©ISTOCKPHOTO.COM/IN-FUTURE

Optimization lies at the heart of machine learning (ML) and signal processing (SP). Contemporary approaches based on the stochastic gradient (SG) method are nonadaptive in the sense that their implementation employs prescribed parameter values that need to be tuned for each application. This article summarizes recent research and motivates future work on adaptive stochastic optimization methods, which have the potential to offer significant computational savings when training large-scale systems.

Introduction

The success of stochastic optimization algorithms for solving problems arising in ML and SP are now widely recognized. Scores of articles have appeared in recent years as researchers aim to build on fundamental methodologies, such as the SG method [26]. The motivation and scope of many of these efforts have been captured in various books and review articles; see, e.g., [4], [10], [15], and [21].

Despite these advances and the accumulation of knowledge, there remain significant challenges in the use of stochastic optimization algorithms in practice. The dirty secret in the use of these algorithms is the tremendous computational costs required to tune them for each application. For large-scale, real-world systems, tuning an algorithm to solve a single problem might require weeks or months of effort on a supercomputer before the algorithm performs well. To appreciate the consumption of energy to accomplish this, the authors of [1] list multiple recent articles in which training a model for a single task requires thousands of CPU days, and they remark how 10^4 CPU days are comparable to driving from Los Angeles to San Francisco with 50 Toyota Camrys. One avenue for avoiding expensive tuning efforts is to employ adaptive optimization algorithms. Long the focus of the deterministic optimization community, with widespread success in practice, such algorithms become significantly more difficult to design for the stochastic regime in which many modern problems reside, including those arising in large-scale ML and SP.

The purpose of this article is to summarize recent work and motivate continued research into the design and analysis of

Digital Object Identifier 10.1109/MSP.2020.3003539
Date of current version: 2 September 2020

1053-5888/20©2020IEEE

adaptive stochastic optimization methods. In particular, we present an analytical framework—new to the literature for adaptive deterministic optimization—that sets the stage for establishing convergence rate guarantees for adaptive stochastic optimization techniques. With this framework in hand, we remark on important open questions related to how the architecture can be extended further for the design of new methods. We also discuss challenges and opportunities for the methods' use in real-world systems.

Background

Many problems in ML and SP are formulated as optimization problems. For example, given a data vector $y \in \mathbb{R}^m$ from an unknown distribution, one often desires to have a vector of model parameters $x \in \mathbb{R}^n$ such that a composite objective function $f : \mathbb{R}^n \to \mathbb{R}$ is minimized, as in

$$\min_{x \in \mathbb{R}^n} f(x), \text{ where } f(x) := \mathbb{E}_y[\phi(x, y)] + \lambda\rho(x).$$

Here, the function $\phi : \mathbb{R}^{n+m} \to \mathbb{R}$ defines the data-fitting term $\mathbb{E}_y[\phi(x, y)]$, an expectation across the distribution of y. For example, in supervised ML, the vector y may represent the input and the output from an unknown mapping, and one aims to find x to minimize the discrepancy between the output vector and the predicted value captured by ϕ. Alternatively, the vector y may represent a noisy signal measurement, and one may aim to find x that filters out the noise to reveal the true signal. The function $\rho : \mathbb{R}^n \to \mathbb{R}$ with the weight $\lambda \in [0, \infty)$ is included as a regularizer. This can be used to induce desirable properties of the vector x, such as sparsity, and/or to help avoid overfitting a particular set of data vectors that is used when (approximately) minimizing f. Supposing that instances of y can be generated—one-by-one or in minibatches, essentially ad infinitum—the problem of minimizing f becomes a stochastic problem across x.

Traditional algorithms for minimizing f are often very simple to understand and implement. For example, given a solution estimate x_k, the well-known and celebrated SG method [26] computes the next estimate as $x_{k+1} \leftarrow x_k - \alpha_k g_k$, where g_k approximates the gradient of f at x_k by taking a uniform random sample y_{i_k} and setting $g_k \leftarrow \nabla_x \phi(x_k, y_{i_k}) + \lambda\nabla_x\rho(x_k)$ (or by taking a minibatch of samples and setting g_k as the average sampled gradient). This value estimates the gradient since, as is typically assumed, $\mathbb{E}_k[g_k] = \nabla f(x_k)$, where $\mathbb{E}_k[\cdot]$ represents conditions on the history of the behavior of the algorithm up to iteration $k \in \mathbb{N}$. Under reasonable assumptions about the stochastic gradient estimates and with a prescribed sequence of step-size parameters $\{\alpha_k\}$, such an algorithm enjoys good convergence properties, which ensure that $\{x_k\}$ converges in probability to a minimizer, or at least a stationary point, of f. For other successful modern variants of SG, see [13] and [18].

A practical issue in the use of SG is that the variance of the stochastic gradient estimates, i.e., $\mathbb{E}_k[\| g_k - \nabla f(x_k) \|_2^2]$, can be large, which inhibits the algorithm from attaining convergence rate guarantees on par with those for first-order algorithms in deterministic settings. To address this, variance reduction techniques have been proposed and analyzed, such as those used in the stochastic variance reduced gradient algorithm, the self-adaptive genetic algorithm, and other methods [12], [17], [23], [27]. That said, SG and its variants are inherently nonadaptive in the sense that each iteration involves a prescribed number of data samples to compute g_k, in addition to a prescribed sequence of step-sizes $\{\alpha_k\}$. Determining which parameter values (defining the minibatch sizes, step-sizes, and other factors) work well for a particular problem is a nontrivial task. Tuning these parameters means that problems cannot be solved once; they need to be solved numerous times until reasonable parameter values are determined for future use on new data.

Illustrative example

To illustrate the use of adaptivity in stochastic optimization, consider a problem of binary classification by logistic regression using the well-known Modified National Institute of Standards and Technology (MNIST) data set. Specifically, consider the minimization of a logistic loss plus an ℓ_2-norm squared regularizer (with $\lambda = 10^{-4}$) to classify images as showing the number five or not. Employing SG with a minibatch size of 64 and different fixed step-sizes, one obtains the plot of the testing accuracy through 10 epochs, as seen in Figure 1(a). One finds that for a step-size of $\alpha_k = 1$ for all $k \in \mathbb{N}$, the model achieves a testing accuracy of roughly 98%. However, for a step-size of $\alpha_k = 0.01$ for all $k \in \mathbb{N}$, the algorithm stagnates and never achieves an accuracy much better than 90%.

By comparison, we also ran an adaptive method. This approach, like SG, begins with a minibatch size of 64 and the step-size parameter indicated in the plot in Figure 1(b). However, in each iteration, it checks the value of the objective (across only the current minibatch) at the current iterate x_k and trial iterate $x_k - \alpha_k g_k$. If the minibatch objective would not reduce sufficiently as a result of the trial step, then the step is not taken, the step-size parameter is reduced by a factor, and the minibatch size is increased by a factor. This results in a more conservative step-size with a more accurate gradient estimate in the subsequent iteration. Otherwise, if the minibatch objective would reduce sufficiently with the trial step, then the step is taken, the step-size parameter is increased by a factor, and the minibatch size is reduced by a factor. Despite the data accesses required by this adaptive algorithm to evaluate minibatch objective values in each iteration, the attained testing accuracy with all initializations competes with that attained by the best SG run.

This experiment demonstrates the potentially significant savings in computational costs offered by adaptive stochastic optimization methods. While one might be able to achieve a good practical performance with a nonadaptive method, one's success might come only after expensive tuning efforts. By contrast, an adaptive algorithm can perform well without such expensive tuning.

Framework for analyzing adaptive deterministic methods

The rigorous development of adaptive stochastic optimization methods requires a solid foundation in terms of convergence rate guarantees. The types of adaptive methods that have enjoyed great success in the realm of deterministic optimization are 1)

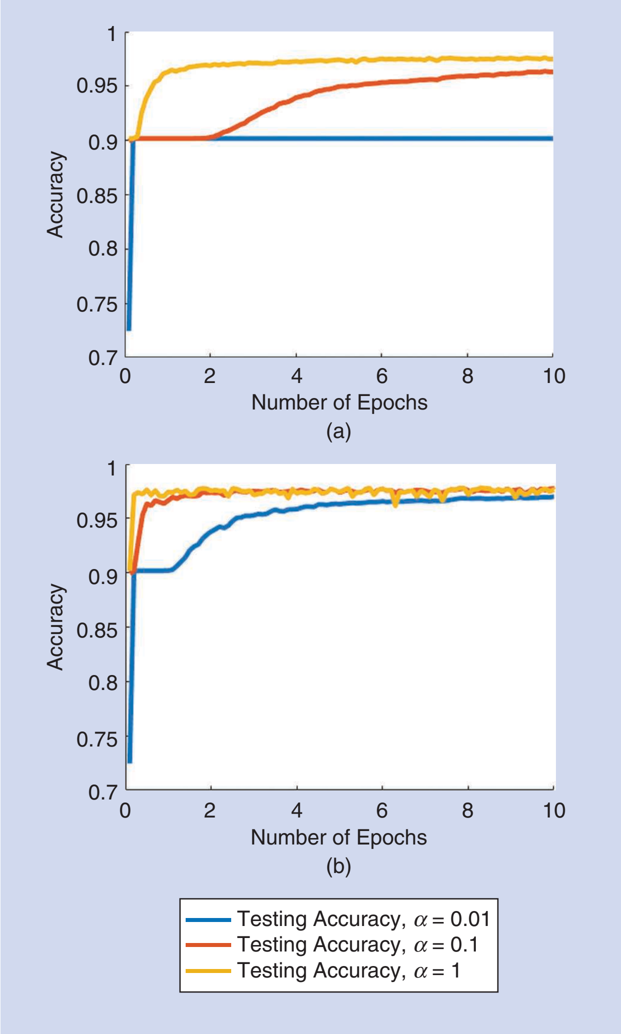


FIGURE 1. SG versus an adaptive stochastic method on regularized logistic regression on the MNIST data set. (a) Accuracies with SG. (b) Accuracies with the adaptive method.

Algorithm 1. Adaptive deterministic framework.

```
Initialization
    Choose constants $\eta \in (0,1), \gamma \in (1,\infty)$, and $\bar{\alpha} \in (0,\infty)$.
    Choose an initial iterate $x_0 \in \mathbb{R}^n$ and step-size parameter
    $\alpha_0 \in (0,\bar{\alpha}]$.
1. Determine model and compute step
    Choose a local model $m_k$ of $f$ around $x_k$. Compute a step $s_k(\alpha_k)$
    such that the model reduction $m_k(x_k) - m_k(x_k + s_k(\alpha_k)) \geq 0$ is
    sufficiently large.
2. Check for sufficient reduction in $f$
    Check if the reduction $f(x_k) - f(x_k + s_k(\alpha_k))$ is sufficiently large
    relative to the model reduction $m_k(x_k) - m_k(x_k + s_k(\alpha_k))$ using a
    condition parameterized by $\eta$.
3. Successful iteration
    If sufficient reduction has been attained (along with other potential
    requirements), then set $x_{k+1} \leftarrow x_k + s_k(\alpha_k)$ and
    $\alpha_{k+1} \leftarrow \min\{\gamma\alpha_k, \bar{\alpha}\}$.
4. Unsuccessful iteration
    Otherwise, $x_{k+1} \leftarrow x_k$, and $\alpha_{k+1} \leftarrow \gamma^{-1}\alpha_k$.
5. Next iteration
    Set $k \leftarrow k+1$.
```

trust region (TR), 2) line search (LS), and 3) regularized Newton methods. These approaches can be applied when derivative estimates are readily available and when using model-based, derivative-free methods, which build gradient and Hessian estimates using function values (see [19] and the references therein). Extending these techniques to the stochastic regime is a highly nontrivial task. After all, these methods traditionally require accurate function information at each iterate, which is what they use to adapt their behavior. When an oracle can return only stochastic function estimates, comparing function values to make adaptive algorithmic decisions can be problematic. In particular, when objective values are merely estimated, poor decisions can be made, and the combined effects of these poor decisions can be difficult to estimate and control.

As a first step toward showing how these challenges can be overcome, let us establish a general framework for convergence analysis for adaptive deterministic optimization. This will lay a foundation for the framework that we present for adaptive stochastic optimization in the "Framework for Analyzing Adaptive Stochastic Methods" section. The analytical framework presented here is new for the deterministic optimization literature. A typical convergence analysis for adaptive deterministic optimization partitions the set of iterations into successful and unsuccessful ones. Nonzero progress in reducing the objective function is made on successful iterations, whereas unsuccessful iterations merely result in an update of a model or algorithmic parameter to promote success in the subsequent iteration. As such, a convergence rate guarantee results from a lower bound on the progress made in each successful iteration and a limit on the number of unsuccessful iterations that can occur between successful ones. By contrast, in the framework presented here, the analysis is structured around a measure in which progress is made in all iterations.

In the remainder of this section, we consider all three aforementioned types of adaptive algorithms under the assumption that f is continuously differentiable, with ∇f being Lipschitz continuous with constant $L \in (0,\infty)$. Each of these methods follows the general algorithmic framework that we state as Algorithm 1. The role of the sequence $\{\alpha_k\} \geq 0$ in the algorithm is to control the length of the trial steps $\{s_k(\alpha_k)\}$. In particular, as seen in our discussion of each type of adaptive algorithm, one presumes that for a given model m_k, the norm of $s_k(\alpha)$ is directly proportional to the magnitude of α. Another assumption—and the reason that we refer to it as a *deterministic optimization framework*—is that the models agree with the objective, at least up to first-order derivatives; i.e., $m_k(x_k) = f(x_k)$ and $\nabla m_k(x_k) = \nabla f(x_k)$ for all $k \in \mathbb{N}$.

Our analysis involves three central ingredients. We define them here in such a way that they are easily generalized when we consider our adaptive stochastic framework later on.

- $\{\Phi_k\} \geq 0$ is a sequence whose role is to measure the progress of the algorithm. The choice of this sequence may vary by the type of algorithm and the assumptions on f.
- $\{W_k\}$ is a sequence of indicators; specifically, for all $k \in \mathbb{N}$, if iteration k is successful, then $W_k = 1$; otherwise, $W_k = -1$.
- T_ε, the stopping time, is the index of the first iterate that satisfies a desired convergence criterion parameterized by ε.

These quantities are not part of the algorithm itself, and therefore do not influence the iterates. They are merely tools of the analysis. At the heart of the analysis is the goal to show that the following condition holds.

Condition 1

The following statements hold with respect to $\{(\Phi_k, \alpha_k, W_k)\}$ and T_ε:

1) There exists a scalar $\underline{\alpha}_\varepsilon \in (0,\infty)$ such that for each $k \in \mathbb{N}$ such that $\alpha_k \le \gamma \underline{\alpha}_\varepsilon$, the iteration is guaranteed to be successful; i.e., $W_k = 1$. Therefore, $\alpha_k \ge \underline{\alpha}_\varepsilon$ for all $k \in \mathbb{N}$.

There exists a nondecreasing function $h_\varepsilon : [0,\infty) \to (0,\infty)$ and a scalar $\Theta \in (0,\infty)$ such that for all $k < T_\epsilon$, $\Phi_k - \Phi_{k+1} \ge \Theta h_\varepsilon(\alpha_k)$.

The goal to satisfy Condition 1 is motivated by the fact that, if the condition holds, it is trivial to derive (since $\Phi_k \ge 0$ for all $k \in \mathbb{N}$) that

$$T_\epsilon \le \frac{\Phi_0}{\Theta h_\varepsilon(\underline{\alpha}_\varepsilon)}. \tag{1}$$

For generality, we have written $\underline{\alpha}_\varepsilon$ and h_ε as parameterized by ε. However, in the context of different algorithms, one or the other of these quantities may be independent of ε. Throughout our analysis, we denote $f_* := \inf_{x \in \mathbb{R}^n} f(x) > -\infty$.

Classical TR

In a classical TR method, the model m_k is chosen as at least a first-order accurate Taylor series approximation of f at x_k, and the step $s_k(\alpha_k)$ is computed as a minimizer of m_k in a ball of radius α_k centered at x_k. In step 2, the sufficient reduction condition is chosen as

$$\frac{f(x_k) - f(x_k + s_k(\alpha_k))}{m(x_k) - m(x_k + s_k(\alpha_k))} \ge \eta. \tag{2}$$

Figure 2 shows the need to distinguish between successful and unsuccessful iterations in a TR method. Even though the model is (at least) first-order accurate, a large TR radius may enable a large enough step such that the reduction predicted by the model does not well represent the reduction in the function itself. We contrast these illustrations later with situations in the stochastic setting that are complicated by the fact that the model might be inaccurate regardless the size of the step.

For simplicity in our discussions, for iteration $k \in \mathbb{N}$ to be successful, we impose the additional condition that $\alpha_k \le \tau \|\nabla f(x_k)\|$ for some suitably large constant $\tau \in (0,\infty)$. (Throughout the article, consider all norms to be ℓ_2.) This condition is actually not necessary for the deterministic setting, but it is needed in our analytical framework in the stochastic setting to ensure that the trial step is not too large compared to the size of the gradient estimate. We impose it now in the deterministic setting for consistency.

For this TR instance of Algorithm 1, consider the first-order ε-stationarity stopping time

$$T_\varepsilon := \min\{k \in \mathbb{N} : \|\nabla f(x_k)\| \le \varepsilon\}, \tag{3}$$

corresponding to which we define

$$\Phi_k := \nu(f(x_k) - f_*) + (1-\nu)\alpha_k^2 \tag{4}$$

for some $\nu \in (0,1)$ (to be determined in the following). Standard TR analysis involves two key results. First, while $\|\nabla f(x_k)\| > \varepsilon$, if α_k is sufficiently small, then iteration k is successful; i.e., $W_k = 1$. In particular, $\alpha_k \ge \underline{\alpha}_\varepsilon := c_1 \varepsilon$ for all $k \in \mathbb{N}$ for some sufficiently small $c_1 \in (0,\infty)$ dependent on L, η, and γ. In addition, if iteration k is successful, then the ratio condition (2) and our imposed condition $\alpha_k \le \tau \|\nabla f(x_k)\|$ yield

$$f(x_k) - f(x_{k+1}) \ge \eta c_2 \alpha_k^2$$

for some $c_2 \in (0,\infty)$, meaning that

$$\Phi_k - \Phi_{k+1} \ge \nu \eta c_2 \alpha_k^2 - (1-\nu)(\gamma^2 - 1)\alpha_k^2;$$

otherwise, if iteration k is unsuccessful, then

$$\Phi_k - \Phi_{k+1} = (1-\nu)(1-\gamma^{-2})\alpha_k^2.$$

We aim to show in either case that $\Phi_k - \Phi_{k+1} \ge \Theta \alpha_k^2$ for some $\Theta > 0$. This can be done by choosing ν sufficiently close to one such that

$$\nu \eta c_2 \ge (1-\nu)(\gamma^2 - \gamma^{-2}).$$

In this manner, it follows from the previous observations that Condition 1 holds with $h_\varepsilon(\alpha_k) := \alpha_k^2$ and $\Theta := (1-\nu)(1-\gamma^{-2})$. Thus, by (1), the number of iterations required until a first-order ε-stationary point is reached satisfies

$$T_\varepsilon \le \frac{\nu(f(x_0) - f_*) + (1-\nu)\alpha_0^2}{(1-\nu)(1-\gamma^{-2})c_1^2 \varepsilon^2}.$$

This shows that $T_\varepsilon = O(\varepsilon^{-2})$.

Classical LS

In a classical LS method, the model is again chosen as at least a first-order accurate Taylor series approximation of f at x_k, with care taken to ensure it is convex so a minimizer of it exists. The

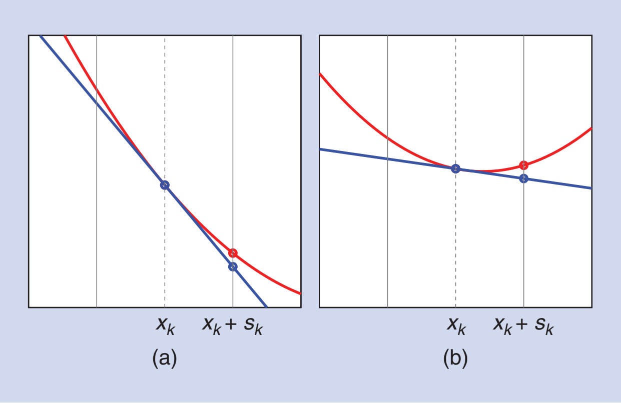


FIGURE 2. (a) Successful and (b) unsuccessful steps in a TR method.

trial step $s_k(\alpha_k)$ is defined as $\alpha_k d_k$ for some direction of sufficient descent d_k. In step 2, the sufficient reduction condition often includes the Armijo condition

$$f(x_k) - f(x_k + s_k(\alpha_k)) \geq -\eta \nabla f(x_k)^T s_k(\alpha_k).$$

As is common, suppose m_k is chosen and d_k is computed such that, for a successful iteration, one finds for some $c_3 \in (0, \infty)$, dependent on L and the angle between d_k and $-\nabla f(x_k)$, that

$$f(x_k) - f(x_{k+1}) \geq \eta c_3 \alpha_k \| \nabla f(x_k) \|^2.$$

Using common techniques, this can be ensured with $\alpha_k \geq \underline{\alpha}$ for all $k \in \mathbb{N}$ for some $\underline{\alpha} \in (0, \infty)$ dependent on L, η, and γ. As in the "Classical TR" section, let us also impose that $\| d_k \| \leq \beta \| \nabla f(x_k) \|$ for some suitably large $\beta \in (0, \infty)$. For this LS instance of Algorithm 1, for the stopping time T_ε defined in (3), consider

$$\Phi_k := \nu(f(x_k) - f_*) + (1 - \nu)\alpha_k \| \nabla f(x_k) \|^2.$$

If iteration k is successful, then

$$\begin{aligned} \Phi_k - \Phi_{k+1} &\geq \nu\eta c_3 \alpha_k \| \nabla f(x_k) \|^2 \\ &-(1-\nu)(\gamma\alpha_k \| \nabla f(x_{k+1}) \|^2 - \alpha_k \| \nabla f(x_k) \|^2). \end{aligned}$$

By the Lipschitz continuity of ∇f, it follows that

$$\| \nabla f(x_{k+1}) \| \leq (L\alpha_k \beta + 1) \| \nabla f(x_k) \|.$$

Squaring both sides and applying $\alpha_k \leq \bar{\alpha}$ yields

$$\| \nabla f(x_{k+1}) \|^2 \leq (L\bar{\alpha}\beta + 1)^2 \| \nabla f(x_k) \|^2.$$

Overall, if iteration k is successful, then

$$\begin{aligned} \Phi_k - \Phi_{k+1} &\geq \nu\eta c_3 \alpha_k \| \nabla f(x_k) \|^2 \\ &-(1-\nu)((L\bar{\alpha}\beta + 1)^2\gamma - 1)\alpha_k \| \nabla f(x_k) \|^2. \end{aligned}$$

On unsuccessful iterations, one simply finds

$$\Phi_k - \Phi_{k+1} \geq (1-\nu)(1-\gamma^{-1})\alpha_k \| \nabla f(x_k) \|^2.$$

By selecting ν sufficiently close to one such that

$$\nu\eta c_3 \geq (1-\nu)((L\bar{\alpha}\beta + 1)^2\gamma - \gamma^{-1}),$$

one ensures, while $\| \nabla f(x_k) \| > \varepsilon$, that Condition 1 holds with $h_\varepsilon(\alpha_k) := \alpha_k \varepsilon^2$, and $\Theta := (1-\nu)(1-\gamma^{-1})$. Thus, by (1), the number of iterations required until a first-order ε-stationary point is reached satisfies

$$T_\varepsilon \leq \frac{\nu(f(x_0) - f_*) + (1-\nu)\alpha_0 \| \nabla f(x_0) \|^2}{(1-\nu)(1-\gamma^{-1})\underline{\alpha}\varepsilon^2},$$

which again shows that $T_\varepsilon = \mathcal{O}(\varepsilon^{-2})$.

Regularized Newton

Regularized Newton methods have been popular during recent years due to their ability to offer optimal worst-case complexity guarantees for nonconvex smooth optimization across the class of practical second-order methods. For example, in the cublicly regularized Newton, the model is chosen as $m_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + (1/2)(x - x_k)^T \nabla^2 f(x_k)(x - x_k) + (1/3)\alpha_k \| x - x_k \|^3$, and in step 2, the imposed sufficient reduction condition has the same form as that in a TR method, namely, the ratio condition (2).

As for the aforementioned LS method, one can show that if the Hessian of f is Lipschitz continuous on a set containing the iterates, an iteration of the cublicly regularized Newton will be successful if the step-size parameter is sufficiently small; consequently, $\alpha_k \geq \underline{\alpha}$ for all $k \in \mathbb{N}$ for some $\underline{\alpha} \in (0, \infty)$ dependent on L, the Lipschitz constant for $\nabla^2 f$, η, and γ. However, to prove the optimal complexity guarantee in this setting, one must account for the progress in a successful iteration as being dependent on the magnitude of the gradient at the next iterate, not the current one. (As we will see, this leads to complications for analysis in the stochastic regime.) For this reason, and ignoring the trivial case when $\| \nabla f(x_0) \| \leq \varepsilon$, let us define

$$T_\varepsilon := \min\{k \in \mathbb{N} : \| \nabla f(x_{k+1}) \| \leq \varepsilon\}$$

along with

$$\Phi_k := \nu(f(x_k) - f_*) + (1-\nu)\alpha_k \| \nabla f(x_k) \|^{\frac{3}{2}}.$$

If iteration k is successful, then

$$f(x_k) - f(x_{k+1}) \geq \eta c_4 \alpha_k \| \nabla f(x_{k+1}) \|^{\frac{3}{2}}$$

for some $c_4 \in (0, \infty)$ [6], meaning that

$$\begin{aligned} \Phi_k - \Phi_{k+1} &\geq \nu\eta c_4 \alpha_k \| \nabla f(x_{k+1}) \|^{\frac{3}{2}} \\ &\quad + (1-\nu)\alpha_k \| \nabla f(x_k) \|^{\frac{3}{2}} \\ &\quad - (1-\nu)\alpha_{k+1} \| \nabla f(x_{k+1}) \|^{\frac{3}{2}} \\ &\geq \nu\eta c_4 \alpha_k \| \nabla f(x_{k+1}) \|^{\frac{3}{2}} \\ &\quad - (1-\nu)\gamma\alpha_k \| \nabla f(x_{k+1}) \|^{\frac{3}{2}}; \end{aligned}$$

otherwise, if iteration k is unsuccessful, then

$$\Phi_k - \Phi_{k+1} \geq (1-\nu)(1-\gamma^{-1})\alpha_k \| \nabla f(x_{k+1}) \|^{\frac{3}{2}}.$$

Choosing ν sufficiently close to one such that

$$\nu\eta c_4 \geq (1-\nu)(\gamma + 1 - \gamma^{-1}),$$

one ensures, while $\| \nabla f(x_{k+1}) \| > \varepsilon$, that Condition 1 holds with $h_\varepsilon(\alpha_k) := \alpha_k \varepsilon^{3/2}$, and $\Theta := (1-\nu_\varepsilon)(1-\gamma^{-1})$. Thus, by (1), the number of iterations required until a first-order ε-stationary point will be reached (in the next iteration) has

$$T_\varepsilon \le \frac{\nu(f(x_0)-f_*)+(1-\nu)\alpha_0\|\nabla f(x_0)\|^{\frac{3}{2}}}{(1-\nu)(1-\gamma^{-1})\underline{\alpha}\varepsilon^{\frac{3}{2}}},$$

which shows that $T_\varepsilon = \mathcal{O}(\varepsilon^{-3/2})$. One can obtain the same result with a second-order TR method; see [9]. It should be said, however, that this method requires a more complicated mechanism for adjusting the step-size parameter than that stated in Algorithm 1.

Additional examples

Our analysis so far has focused on the setting of having a stopping time based on a first-order stationarity criterion and no assumptions on f besides first- and/or second-order Lipschitz continuous differentiability. However, the framework can be extended to other situations, as well. For example, if one is interested in approximate second-order stationarity, then one can let

$$T_\varepsilon := \min\{k \in \mathbb{N} : \chi_k \le \varepsilon\},$$
$$\text{where } \chi_k := \max\{\|\nabla f(x_k)\|, -\lambda_{\min}(\nabla^2 f(x_k))\},$$

with $\lambda_{\min}(\cdot)$ denoting the minimum eigenvalue of its symmetric matrix argument. One can show that if the model m_k is chosen as a (regularized) second-order Taylor series approximation of f at x_k, then for all the aforementioned methods, one obtains $T_\varepsilon = \mathcal{O}(\varepsilon^{-3})$. For a TR method, for example, one can derive this using

$$\Phi_k := \nu(f(x_k)-f_*)+(1-\nu)\alpha_k^3.$$

On the other hand, if one is interested in specially analyzing the case of f being convex, or even strongly convex, then one might consider

$$T_\varepsilon := \min\{k \in \mathbb{N} : f(x_k)-f_* \le \varepsilon\}.$$

In this case, improved complexity bounds can be obtained through other careful choices of $\{\Phi_k\}$. For example, when f is convex and the LS method from the "Classical LS" section is employed, one can let

$$\Phi_k := \nu\left(\frac{1}{\varepsilon}-\frac{1}{f(x_k)-f_*}\right)+(1-\nu)\alpha_k.$$

Under the assumption that level sets of f are bounded, one can show that $(f(x_{k+1})-f_*)^{-1}-(f(x_k)-f_*)^{-1}$ is uniformly bounded below by a positive constant across all successful iterations, while for all $k \in \mathbb{N}$, one has $\alpha_k \ge \underline{\alpha}$. Hence, for a suitable constant ν, one can determine $h_\varepsilon(\alpha_k)$ and Θ to satisfy Condition 1. In this case, the function h_ε does not depend on ε but on $\Phi_0 = \mathcal{O}(\varepsilon^{-1})$, meaning that $T_\varepsilon = \mathcal{O}(\varepsilon^{-1})$.

Similarly, when f is strongly convex and the LS method from the "Classical LS" section is employed, consider

$$\Phi_k := \nu\left(\log\left(\frac{1}{\varepsilon}\right)-\log\left(\frac{1}{f(x_k)-f_*}\right)\right)$$
$$+(1-\nu)\log(\alpha_k).$$

This time, $\log((f(x_{k+1})-f_*)^{-1})-\log((f(x_k)-f_*)^{-1})$ is uniformly bounded below by a positive constant across all successful iterations, and, similar to the convex case, one can determine ν, h_ε, and Θ independent of ε to show that $\Phi_0 = \mathcal{O}(\log(\varepsilon^{-1}))$ implies $T_\varepsilon = \mathcal{O}(\log(\varepsilon^{-1}))$.

Framework for analyzing adaptive stochastic methods

We now present a generalization of the framework introduced in the previous section that facilitates the analysis of adaptive stochastic optimization methods. This framework is based on the techniques proposed in [7], which served to analyze the behavior of algorithms when derivative estimates are stochastic and function values can be computed exactly. It is also based on the subsequent work in [2], which enables function values to be stochastic, as well. For our purposes of providing intuition, we discuss a simplification of the framework, avoiding some technical details. See [2] and [7] for complete information.

As in the deterministic setting, let us define a generic algorithmic framework, which we state as Algorithm 2, that encapsulates multiple types of adaptive algorithms. This algorithm has the same structure as Algorithm 1, except it makes use of a stochastic model of f to compute the trial step, and it employs stochastic objective value estimates when determining whether a sufficient reduction has been achieved.

Corresponding to Algorithm 2, let $\{(\Phi_k, W_k)\}$ be a stochastic process such that $\{\Phi_k\} \ge 0$ for all $k \in \mathbb{N}$. The sequences $\{\Phi_k\}$ and $\{W_k\}$ play roles similar to those in our analysis of Algorithm 1, but it should be noted that now each W_k is a random indicator influenced by the iterate sequence $\{x_k\}$ and the step-size parameter sequence $\{\alpha_k\}$, which are themselves stochastic processes.

Let $\mathcal{F}_k$ denote the σ-algebra generated by all the stochastic processes within Algorithm 1 at the beginning of iteration k. Roughly, $\mathcal{F}_k$ is generated by $\{(\Phi_j, \alpha_j, x_j)\}_{j=0}^{k}$. Note that this

Algorithm 2. Adaptive stochastic framework.

Initialization
Choose $(\eta, \delta_1, \delta_2) \in (0,1)^3$, $\gamma \in (1,\infty)$, and $\bar{\alpha} \in (0,\infty)$.
Choose an initial iterate $x_0 \in \mathbb{R}^n$ and step-size parameter $\alpha_0 \in (0, \bar{\alpha}]$.

1. Determine model and compute step
Choose a stochastic model m_k of f around x_k, which satisfies some sufficient accuracy requirement with a probability of at least $1-\delta_1$. Compute a step $s_k(\alpha_k)$ such that the model reduction $m_k(x_k)-m_k(x_k+s_k(\alpha_k)) \ge 0$ is sufficiently large.

2. Check for sufficient reduction
Compute estimates $\tilde{f}_k^0$ and $\tilde{f}_k^s$ of $f(x_k)$ and $f(x_k+s_k(\alpha_k))$, respectively, which satisfy some sufficient accuracy requirement with a probability of at least $1-\delta_2$. Check if $\tilde{f}_k^0-\tilde{f}_k^s$ is sufficiently large relative to the model reduction $m_k(x_k)-m_k(x_k+s_k(\alpha_k))$ using a condition parameterized by η.

3. Successful iteration
If a sufficient reduction has been attained (along with other potential requirements), then set $x_{k+1} \leftarrow x_k+s_k(\alpha_k)$ and $\alpha_{k+1} \leftarrow \min\{\gamma\alpha_k, \bar{\alpha}\}$.

4. Unsuccessful iteration
Otherwise, $x_{k+1} \leftarrow x_k$, and $\alpha_{k+1} \leftarrow \gamma^{-1}\alpha_k$.

5. Next iteration
Set $k \leftarrow k+1$.

includes $\{(\Phi_j, \alpha_j, x_j)\}_{j=0}^{k-1}$ but not W_k, as this random variable depends on what happens during iteration k. We then let T_ε denote a family of stopping times for $\{(\Phi_k, W_k)\}$ with respect to $\{\mathcal{F}_k\}$ parameterized by $\varepsilon \in (0, \infty)$. The goal of our analytical framework is to derive an upper bound for the expected stopping time $\mathbb{E}[T_\varepsilon]$ under various assumptions on the behavior of $\{(\Phi_k, W_k)\}$ and on the objective f. At the heart of the analysis is the goal to show that the following condition holds, which can be seen as a generalization of Condition 1.

Condition 2

The following statements hold with respect to $\{(\Phi_k, \alpha_k, W_k)\}$ and T_ε.

1) There exists a scalar $\underline{\alpha}_\varepsilon \in (0, \infty)$ such that, conditioned on the event that $\alpha_k \le \underline{\alpha}_\varepsilon$, one has $W_k = 1$ with the probability $1 - \delta > 1/2$, conditioned on $\mathcal{F}_k$. (This means that, if it becomes sufficiently small, one is more likely to see an increase in the step-size parameter than a decrease in it.)
2) There exists a nondecreasing function $h_\varepsilon : [0, \infty) \to (0, \infty)$ and a scalar $\Theta \in (0, \infty)$ such that, for all $k < T_\varepsilon$, the conditional expectation of $\Phi_k - \Phi_{k+1}$ with respect to $\mathcal{F}_k$ is at least $\Theta h_\varepsilon(\alpha_k)$; specifically,

$$\mathbb{1}_{\{k<T_\varepsilon\}}\mathbb{E}[\Phi_{k+1} \mid \mathcal{F}_k] \le \mathbb{1}_{\{k<T_\varepsilon\}}(\Phi_k - \Theta h_\varepsilon(\alpha_k)).$$

Whereas Condition 1 requires that the step-size parameter α_k remains above a lower bound and that the reduction $\Phi_k - \Phi_{k+1}$ is nonnegative with certainty, Condition 2 allows more flexibility. In particular, it says that until the stopping time is reached, one tends to find $\Phi_k - \Phi_{k+1}$ sufficiently large, namely, at least $\Theta h_\varepsilon(\alpha_k)$ for each $k < T_\varepsilon$. In addition, the step-size parameter is allowed to fall below the threshold $\underline{\alpha}_\varepsilon$; in fact, it can become arbitrarily small. That said, if $\alpha_k \le \underline{\alpha}_\varepsilon$, then one tends to find $\alpha_{k+1} > \alpha_k$. With this added flexibility, one can still prove complexity guarantees. Intuitively, the reduction $\Phi_k - \Phi_{k+1}$ is at least the deterministic amount $\Theta h_\varepsilon(\underline{\alpha}_\varepsilon)$ often enough that one is able to bound the total number of such occurrences (since $\{\Phi_k\} \ge 0$). The following theorem (see [2, Th. 2.2]) bounds the expected stopping time in terms of a deterministic value.

Theorem 1

If Condition 2 holds, then

$$\mathbb{E}[T_\varepsilon] \le \frac{1-\delta}{1-2\delta} \cdot \frac{\Phi_0}{\Theta h_\varepsilon(\underline{\alpha}_\varepsilon)} + 1.$$

In the "Stochastic TR" and "Stochastic LS" sections, we summarize how this framework has been applied to analyze the behavior of stochastic TR and LS methods. In each case, the keys to applying the framework are determining how to design the process $\{\Phi_k\}$ as well as how to specify details of Algorithm 2 to ensure that Condition 2 holds. For these aspects, we first need to describe different adaptive accuracy requirements for stochastic functions and derivative estimates that might be imposed in steps 1 and 2 as well the techniques that have been developed to ensure these requirements.

Error bounds for stochastic functions and derivative estimates

In this section, we describe various types of conditions that one may require in an adaptive stochastic optimization algorithm when computing the objective function, gradient, and Hessian estimates. These conditions have been used in some previously proposed adaptive stochastic optimization algorithms [2], [5], [7], [24].

Let us remark in passing that one does not necessarily need to employ sophisticated error bound conditions in stochastic optimization to achieve improved convergence rates. For example, in the case of minimizing the strongly convex f, the linear rate of convergence of the gradient descent can be emulated by an SG-like method if the minibatch size grows exponentially [14], [25]. However, attaining similar improvements in the (not strongly) convex and nonconvex settings has proved elusive. Moreover, while the stochastic estimates become better with the progress of such an algorithm, this improvement is based on prescribed parameters, and hence the algorithm is not adaptive in our desirable sense. Therefore, one still needs to tune such algorithms for each application. Returning to our setting, in the types of error bounds presented in the following, for a given $f : \mathbb{R}^n \to \mathbb{R}$ and $x \in \mathbb{R}^n$, let $\tilde{f}(x)$, $g(x)$, and $H(x)$ denote stochastic approximations of $f(x)$, $\nabla f(x)$, and $\nabla^2 f(x)$, respectively.

Taylor-like conditions

Corresponding to a norm $\|\cdot\|$, let $\mathbb{B}(x_k, \Delta_k)$ denote a ball of radius Δ_k centered at x_k. If the function, gradient, and Hessian estimates satisfy

$$\left|\tilde{f}(x_k) - f(x_k)\right| \le \kappa_f \Delta_k^2, \tag{5a}$$

$$\left\| g(x_k) - \nabla f(x_k) \right\| \le \kappa_g \Delta_k, \tag{5b}$$

and

$$\left\| H(x_k) - \nabla^2 f(x_k) \right\| \le \kappa_H \tag{5c}$$

for some nonnegative scalars $(\kappa_f, \kappa_g, \kappa_H)$, then the model $m_k(x) = \tilde{f}(x_k) + g(x_k)^T(x - x_k) + (1/2)(x - x_k)^T H(x_k)(x - x_k)$ gives an approximation of f within $\mathbb{B}(x_k, \Delta_k)$ that is comparable to that given by an accurate first-order Taylor series approximation (with error dependent on Δ_k). Similarly, if (5) holds with the right-hand side values replaced by $\kappa_f \Delta_k^3$, $\kappa_g \Delta_k^2$, and $\kappa_H \Delta_k$, respectively, then m_k gives an approximation of f that is comparable to that given by an accurate second-order Taylor series approximation. In a stochastic setting, when unbiased estimates of $f(x_k)$, $\nabla f(x_k)$, and $\nabla^2 f(x_k)$ can be computed, such conditions can be ensured, with some sufficiently high probability of $1 - \delta$, by sample average approximations using a sufficiently large number of samples. For example, to satisfy (5b) with the probability $1 - \delta$, the sample size for computing $g(x_k)$ can be $\Omega(V_g/(\kappa_g^2 \Delta_k^2))$, where V_g is the variance of the stochastic gradient estimators corresponding to a minibatch size of one. Here, the Ω notation hides the dependence on δ, which is weak if $\| g(x_k) - \nabla f(x_k) \|$ is bounded.

Gradient norm condition

If, for some $\theta \in [0, 1)$, one has

$$\left\| g(x_k) - \nabla f(x_k) \right\| \le \theta \left\| \nabla f(x_k) \right\|, \tag{6}$$

then we say that $g(x_k)$ satisfies the gradient norm condition at x_k. Unfortunately, verifying the gradient norm condition at x_k requires knowledge of $\|\nabla f(x_k)\|$, which makes it an impractical condition. In [5], a heuristic is proposed that attempts to approximate the sample size for which the gradient norm condition holds. More recently, in [3], the authors improve on the gradient norm condition by introducing an angle condition, which, in principle, enables smaller sample set sizes to be employed. However, again, the angle condition requires a bound in terms of $\|\nabla f(x_k)\|$, for which a heuristic estimate needs to be employed.

Instead of employing the unknown quantity $\|\nabla f(x_k)\|$ on the right-hand side of the norm and the angle conditions, one can substitute $\varepsilon \in (0, \infty)$, the desired stationarity tolerance. In this manner, while $\|\nabla f(x_k)\| > \varepsilon$ and $\|g_k - \nabla f(x_k)\| \leq \theta\varepsilon$, the norm condition is satisfied. This general idea has recently been exploited in several articles (e.g., [28] and [30]), where gradient and Hessian estimates are computed based on large-enough numbers of samples, then assumed to be accurate in every iteration (with a high probability) until an ϵ-stationary solution is reached. While strong iteration complexity guarantees can be proved for such algorithms, these approaches are too conservative to be competitive with truly stochastic algorithms.

Stochastic gradient norm conditions

Consider again the conditions in (5), although now contemplate the specific setting of defining $\Delta_k := \alpha_k \|g(x_k)\|$ for all $k \in \mathbb{N}$. This condition, which involves a bound comparable to (6) when $g(x_k) = \nabla f(x_k)$, is particularly useful in the context of LS methods. It is possible to impose it in probability since, unlike (6), it does not require knowledge of $\|\nabla f(x_k)\|$. In this case, to satisfy (5) for $\Delta_k := \alpha_k \|g(x_k)\|$ with a probability of $1-\delta$, the sample size for computing $g(x_k)$ need only be $\Omega(V_g/(\kappa_g^2 \alpha_k^2 \|g(x_k)\|^2))$. While $g(x_k)$ is not known when the sample size for computing it is chosen, one can use a simple loop that guesses the value of $\|g(x_k)\|$, then iteratively increases the number of samples as needed; see [7] and [24].

Note that all of the preceding conditions can be adaptive in terms of the progress of an algorithm, assuming that $\{\Delta_k\}$ and/or $\{\|g(x_k)\|\}$ vanish as $k \to \infty$. However, this behavior does not have to be monotonic, which is a benefit of adaptive step-sizes and accuracy requirements.

Stochastic TR

The idea of designing stochastic TR methods has been attractive for years, even before the recent explosion of efforts on algorithms for stochastic optimization. This is due to the impressive practical performance that TR methods offer, especially when (approximate) second-order information is available. Since TR methods typically compute trial steps by minimizing a quadratic model of the objective in a neighborhood around the current iterate, they are inherently equipped to avoid certain spurious stationary points that are not local minimizers. In addition, by normalizing the step length by a TR radius, the behavior of the algorithm is kept relatively stable. Indeed, this feature enables TR methods to offer stability, even in the nonadaptive stochastic regime; see [11].

However, it has not been until the last couple of years that researchers have been able to design stochastic TR methods that can offer strong expected complexity bounds, which are essential for ensuring that the practical performance of such methods can be competitive across broad classes of problems; see, e.g., [16]. The recently proposed algorithm known as *stochastic optimization with random models* (*STORM*), introduced in [8], achieves such complexity guarantees by requiring the Taylor-like conditions (5) to hold with a probability of at least $1-\delta$, conditioned on $\mathcal{F}_k$. (In what follows, all accuracy conditions are assumed to hold with some probability, conditioned on $\mathcal{F}_k$. We omit the mention of this conditioning for brevity.) In particular, in step 1 of Algorithm 2, a model $m_k(x) := \tilde{f}(x_k) + g(x_k)^T(x - x_k) + (1/2)(x - x_k)^T H(x_k)(x - x_k)$ is computed with components satisfying (5) with a probability of $1-\delta_1$ (where $\Delta_k := \alpha_k$ for all $k \in \mathbb{N}$), then the step $s_k(\alpha_k)$ is computed by minimizing m_k (approximately) within a ball of radius α_k. In step 2, the estimates $\tilde{f}_k^0$ and $\tilde{f}_k^s$ are computed to satisfy (5a) with a probability of $1-\delta_2$. The imposed sufficient reduction condition is

$$\frac{\tilde{f}_k^0 - \tilde{f}_k^s}{m_k(x_k) - m_k(x_k + s_k)} \geq \eta.$$

An iteration is successful if the preceding holds and $\|g_k\| \geq \tau\alpha_k$ for some user-defined $\tau \in (0, \infty)$.

For brevity, we omit some details of the algorithm. For example, the constant κ_f in (5) cannot be too large, whereas κ_g and κ_H can be arbitrarily large. Naturally, the magnitudes of these constants affect the constants in the convergence rate; see [2] for further details. Based on the stochastic process generated by Algorithm 2, let us define, as in the deterministic setting, T_ε and $\{\Phi_k\}$ by (3) and (4), respectively. It is shown in [2] that for sufficiently small constants δ_1, δ_2, and Θ (independent of ε), Condition 2 holds with

$$\underline{\alpha}_\varepsilon := \zeta\varepsilon \text{ and } h(\alpha_k) := \alpha_k^2$$

for some positive constants ζ and Θ that depend on the algorithm parameters and properties of f but not on ε. The probability $1-\delta$ that arises in Condition 2 is at least $(1-\delta_1)(1-\delta_2)$. Thus, by Theorem 1, the expected complexity of reaching a first-order ε-stationary point is at most $\mathcal{O}(\varepsilon^{-2})$, which matches the complexity of the deterministic version of the algorithm up to the factor dependent on $(1-\delta_1)(1-\delta_2)$.

Let us give some intuition of how Condition 2 is ensured. Let us say that the model m_k is "good" if its components satisfy (5) and "bad" otherwise. Similarly, the estimates f_k^0 and f_k^s are "good" if they satisfy (5a) and "bad" otherwise. By the construction of steps 1 and 2 of the algorithm, m_k is "good," with a probability of at least $1-\delta_1$, and f_k^0 and f_k^s are "good," with a probability of at least $1-\delta_2$. Figure 3 illustrates the four possible outcomes. When both the model and the estimates are good, the algorithm essentially behaves as its deterministic counterpart; in particular, if $\alpha_k \leq \underline{\alpha}_\varepsilon$ and $k < T_\varepsilon$, then the kth iteration is successful, and the reduction $\Phi_k - \Phi_{k+1}$ is sufficiently large. If the model is bad and the estimates are good, or if the model is good and the estimates are bad, then the worst case (depicted in Figure 3) is that the step is deemed unsuccessful, even though $\alpha_k \le \underline{\alpha}_\varepsilon$. This shows that the step-size parameter can continue to decrease, even if it is already small. Finally, if both the model and the estimates are bad, which happens with a probability of at most $\delta_1\delta_2$, then it is possible that the iteration will be deemed successful despite the fact that $\Phi_{k+1} > \Phi_k$. (Recall that this cannot occur in the deterministic setting, where $\{\Phi_k\}$ decreases monotonically.) The key step in showing that $\Phi_{k+1} < \Phi_k$ in expectation is to establish that, in each iteration, the possible decrease of this measure is proportional to any possible increase, and thus by ensuring that $\delta_1\delta_2$ is sufficiently small and $(1-\delta_1)(1-\delta_2)$ is sufficiently large, one can guarantee a desired reduction in expectation.

The same TR algorithm can be employed with minor modifications to obtain good expected complexity properties with respect to achieving second-order ε-stationarity. In this case, the requirements on the estimates need to be stronger; in particular, (5) has to be imposed with right-hand side values $\kappa_f\alpha_k^3$, $\kappa_g\alpha_k^2$, and $\kappa_H\alpha_k$, respectively. In this case, Condition 2 holds for $h(\alpha_k) = \alpha_k^3$. Hence, the expected complexity of reaching a second-order ε-stationary point by Algorithm 2 is bounded by $O(\varepsilon^{-3})$, which similarly matches the deterministic complexity.

Stochastic LS

A major disadvantage of an algorithm such as SG is that one is very limited in the choice of step-size sequences that can be employed to adhere to the theoretical guidelines. One would like to be able to employ a type of line search, as has been standard practice throughout the history of research on deterministic optimization algorithms. However, devising LS methods for the stochastic regime turns out to be extremely difficult. This is partially due to the fact that, unlike TR methods, LS algorithms employ steps that are highly influenced by the norm of the gradient $\|\nabla f(x_k)\|$, or in the stochastic regime, influenced by $\|g(x_k)\|$. Since the norm of the gradient estimate may have a high variance, the algorithm needs to have a carefully controlled step-size selection mechanism to ensure convergence.

In [3] and [14], two backtracking LS methods were proposed that use different heuristic sample size strategies when computing gradient and function estimates. In both cases, the backtracking is based on the Armijo condition applied to function estimates that are computed on the same batch as the gradient estimates. A different type of LS method that uses a probabilistic Wolfe condition for choosing the step-size was proposed in [20], although this approach possesses no known theoretical guarantees.

In [29], the authors argue that with the overparameterization of deep neural networks (DNNs), the variance of stochastic gradients tends to zero near stationarity points, and they analyze a stochastic LS method under this assumption. This supposition makes the design and analysis of stochastic methods significantly easier, and it might also be used to simplify the methods considered in this article. However, it is not generally true, even for DNNs; thus, our focus is on methods that adaptively control the variance.

Here, we summarize the results in [24], where an LS method with an adaptive sample size selection mechanism is proposed and complexity bounds are provided. This method can be described as a particular case of Algorithm 2. As in the deterministic case, a stochastic model m_k is chosen in step 1, and $s(\alpha_k) = -\alpha_k d_k$, where d_k makes an obtuse angle with the gradient estimate $g(x_k)$. The sufficient reduction in step 2 is based on the estimated Armijo condition

$$\tilde{f}_k^0 - \tilde{f}_k^s \ge -\eta g(x_k)^T s_k(\alpha_k).$$

The algorithm requires that the components of the model m_k satisfy (5) with a probability of at least $1-\delta_1$ and that the estimates $\tilde{f}_k^0$ and $\tilde{f}_k^s$ satisfy (5a) with a probability of at least $1-\delta_2$. Here, it is critical that (5a) not be imposed with the right-hand side being $\alpha_k\|g_k\|^2$ (even though the deterministic case might suggest this as being appropriate) since this quantity can vary uncontrollably from one iteration to the next. To avoid this issue, the approach defines an additional control sequence $\{\Delta_k\}$ used for governing the accuracy of $\tilde{f}_k^0$ and $\tilde{f}_k^s$. Intuitively, for all $k \in \mathbb{N}$, the value Δ_k^2 is meant to approximate $\alpha_k\|\nabla f(x_k)\|^2$, which, as seen in the deterministic case, is the desired reduction in f if iteration k is successful. This control sequence needs to be set carefully. The first value in the sequence is set arbitrarily, with subsequent values set as follows. If iteration k is unsuccessful,

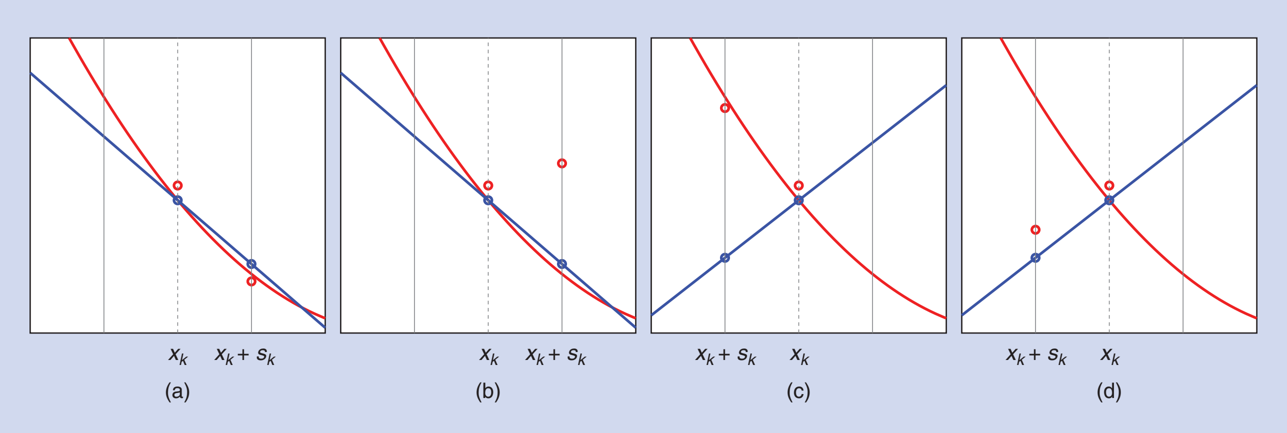


FIGURE 3. “Good” and “bad” models and estimates in a stochastic TR method. (a) A good model and good estimates. (b) A good model and bad estimates. (c) A bad model and good estimates. (d) A bad model and bad estimates.

then $\Delta_{k+1} \leftarrow \Delta_k$. Otherwise, if iteration k is successful, then one checks whether the step is reliable in the sense that the accuracy parameter is sufficiently small; i.e.,

$$\alpha_k \| g(x_k) \|^2 \geq \Delta_k^2. \tag{7}$$

If (7) holds, then one sets $\Delta_{k+1} \leftarrow \sqrt{\gamma}\,\Delta_k$; otherwise, one sets $\Delta_{k+1} \leftarrow \sqrt{\gamma^{-1}}\,\Delta_k$ to promote the reliability of the step in the subsequent iteration. Using $\{\Delta_k\}$ defined in this manner, an additional bound is imposed on the variance of the objective value estimates. For all $k \in \mathbb{N}$, one requires

$$\begin{aligned}&\max\left\{\mathbb{E}\left|\tilde{f}_k^0 - f(x_k)\right|^2, \mathbb{E}\left|\tilde{f}_k^s - f(x_k + s_k(\alpha_k))\right|^2\right\} \\ &\quad\leq \max\left\{\kappa_f \alpha_k^2 \| \nabla f(x_k) \|^2, \kappa_f \Delta_k^4\right\}.\end{aligned}$$

Note that because of the "max" in the right-hand side of this inequality, and because Δ_k is a known value, it is not necessary to know $\| \nabla f(x_k) \|^2$ to impose this condition. Also note that this condition is stronger than imposing the Taylor-like condition (5a) with some probability less than one because the bound on expectation does not allow $|\tilde{f}_k^0 - f(x_k)|$ to be arbitrarily large with a positive probability, while (5a) permits it and thus is more tolerant to outliers.

For analyzing this LS instance of Algorithm 2, again let T_ε be as in (3). However, now let

$$\Phi_k := \nu(f(x_k) - f_*) + (1-\nu)\left(\frac{\alpha_k}{L^2}\| \nabla f(x_k) \|^2 + \eta \Delta_k^2\right).$$

Using a strategy similar to that for STORM combined with the logic of the "Classical LS" section, it is shown in [24] that Condition 2 holds with

$$h(\alpha_k) = \alpha_k \varepsilon^2.$$

The expected complexity of this stochastic LS method has been analyzed for minimizing convex and strongly convex f using modified definitions for Φ_k and T_ε, as described in the "Additional Examples" section; see [24].

Stochastic regularized Newton

As we have seen, stochastic TR and LS algorithms have been developed that fit into the adaptive stochastic framework that we have described, showing that they can achieve expected complexity guarantees on par with their deterministic counterparts. However, neither of these types of algorithms achieves complexity guarantees that are optimal in the deterministic regime.

Cublicly regularized Newton [6], [22], described and analyzed in the "Regularized Newton" section, enjoys optimal convergence rates for second-order methods for minimizing nonconvex functions. We have shown how our adaptive deterministic framework gives a $O(\varepsilon^{-3/2})$ complexity bound for this method for achieving first-order ε-stationarity, in particular, to achieve $\| \nabla f(x_{k+1}) \| \leq \varepsilon$. There have also been works that propose stochastic and randomized versions of cubic regularization methods [28], [30], but these impose strong conditions on the accuracy of the function, gradient, and Hessian estimates that are equivalent to using ε (i.e., the desired accuracy threshold) in place of Δ_k for all $k \in \mathbb{N}$ in (5). Thus, these approaches essentially reduce to sample average approximation with very tight accuracy tolerances and are not adaptive enough to be competitive with truly stochastic algorithms in practice. For example, in [28], no adaptive step-sizes or batch sizes are employed, and in [30], only Hessian approximations are assumed to be stochastic. Moreover, the convergence analysis is performed under the assumption that the estimates are sufficiently accurate (for a given ε) in every iteration. Thus, essentially, the analysis is reduced to that in the deterministic setting and applies only as long as no iteration fails to satisfy the accuracy condition. Hence, a critical open question is whether one can extend the framework described here (or another approach) to develop and analyze a stochastic algorithm that achieves optimal deterministic complexity.

The key difficulty in extending the analysis described in the "Regularized Newton" section to the stochastic regime is the definition of the stopping time. For Theorem 1 to hold, T_ε has to be a stopping time with respect to $\{\mathcal{F}_k\}$. However, with $s_k(\alpha_k)$ being random, x_{k+1} is not measurable in $\{\mathcal{F}_k\}$; hence, T_ε, as it is defined in the deterministic setting, is not a valid stopping time in the stochastic regime. A different definition is needed that would be agreeable with the analysis in the stochastic setting.

Another algorithmic framework that enjoys optimal complexity guarantees is the TR Algorithm with Contractions and Expansions (TRACE) [9]. This algorithm borrows much from the traditional TR methodology, which is also followed by STORM, but it incorporates a few algorithmic variations that reduce the complexity from $O(\varepsilon^{-2})$ to $O(\varepsilon^{-3/2})$ for achieving first-order ε-stationarity. It remains an open question whether one can employ the adaptive stochastic framework to analyze a stochastic variant of TRACE, the main challenge being that TRACE involves a relatively complicated strategy for updating the step-size parameter. Specifically, deterministic TRACE requires knowledge of the exact Lagrange multiplier of the TR constraint at a solution of the step computation subproblem. If the model m_k is stochastic and the subproblem is solved only approximately, then it remains open how to maintain the optimal (expected) complexity guarantee. In addition, the issue of determining the correct stopping time in the stochastic regime is also open for this method.

Other possible extensions

We have shown that the analytical framework for analyzing adaptive stochastic optimization algorithms presented in the "Framework for Analyzing Adaptive Stochastic Methods" section has offered a solid foundation on which stochastic TR and stochastic LS algorithms have been proposed and analyzed. We have also shown the challenges and opportunities for extending the use of this framework for analyzing algorithms whose deterministic counterparts have optimal complexity.

Other interesting questions remain to be answered. For example, great opportunities exist for the design of error bound conditions other than those mentioned in the "Error Bounds for Stochastic Function and Derivative Estimates" section, especially when it comes to bounds that are tailored for particular problem settings. While improved error bounds might not lead to advances in the iteration complexity, they can have great effects on the work complexity of various algorithms, which translates directly into performance gains in practice.

The proposed analytical framework might also benefit from extensions in terms of the employed algorithmic parameters. For example, rather than using a single constant γ when updating the step-size parameter, one might consider different values for increases versus decreases as well as when different types of steps are computed. This will enable improved bounds on the accuracy probability tolerances δ_1 and δ_2.

Finally, numerous open questions remain in terms of how best to implement adaptive stochastic algorithms in practice. For different algorithm instances, practitioners need to explore how best to adjust minibatch sizes and step-sizes so that one can truly achieve good performance without wasteful tuning efforts. One hopes that with additional theoretical advances, these practical questions will become easier to answer.

Authors

Frank E. Curtis (frank.e.curtis@gmail.com) received his B.S. degree from the College of William and Mary and his M.S. and Ph.D. degrees from Northwestern University. He was a postdoctoral researcher at the Courant Institute and is now an associate professor in the Department of Industrial and Systems Engineering, Lehigh University. He is a recipient of the INFORMS Computing Society Prize. His research focuses on the design, analysis, and implementation of algorithms for large-scale nonlinear optimization problems.

Katya Scheinberg (katyas@cornell.edu) received her B.S. degree from Moscow State University and her Ph.D. degree from Columbia University. She is a professor at the School of Operations Research and Information Engineering, Cornell University. She is the coauthor of *Introduction to Derivative Free Optimization* with Andrew R. Conn and Luis N. Vicente, for which they were awarded the Lagrange Prize in Continuous Optimization. Her main research areas are related to developing practical algorithms (and their theoretical analysis) for various problems in continuous optimization, such as convex optimization, derivative-free optimization, machine learning, quadratic programming, and so forth. Her recent research focuses on the analysis of probabilistic methods and stochastic optimization with a variety of applications in machine learning and reinforcement learning. In 2019, she was awarded the Farkas Prize by the INFORMS Optimization Society.

References

[1] H. Asi and J. C. Duchi, "The importance of better models in stochastic optimization," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 116, no. 46, pp. 22,924–22,930, 2019. doi: 10.1073/pnas.1908018116.

[2] J. Blanchet, C. Cartis, M. Menickelly, and K. Scheinberg, "Convergence rate analysis of a stochastic trust-region method via supermartingales," *INFORMS J. Optim.*, vol. 1, no. 2, pp. 92–119, 2019. doi: 10.1287/ijoo.2019.0016.

[3] R. Bollapragada, R. Byrd, and J. Nocedal, "Adaptive sampling strategies for stochastic optimization," *SIAM J. Optim.*, vol. 28, no. 4, pp. 3312–3343, 2018. doi: 10.1137/17M1154679.

[4] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Rev.*, vol. 60, no. 2, pp. 223–311, 2018. doi: 10.1137/16M1080173.

[5] R. Byrd, G. M. Chin, J. Nocedal, and Y. Wu, "Sample size selection in optimization methods for machine learning," *Math. Program.*, vol. 134, no. 1, pp. 127–155, 2012. doi: 10.1007/s10107-012-0572-5.

[6] C. Cartis, N. I. M. Gould, and P. L. Toint, "Adaptive cubic regularisation methods for unconstrained optimization. Part II: Worst-case function- and derivative-evaluation complexity," *Math. Program.*, vol. 130, no. 2, pp. 295–319, 2011. doi: 10.1007/s10107-009-0337-y.

[7] C. Cartis and K. Scheinberg, "Global convergence rate analysis of unconstrained optimization methods based on probabilistic models," *Math. Program.*, vol. 169, no. 2, pp. 337–375, 2018. doi: 10.1007/s10107-017-1137-4.

[8] R. Chen, M. Menickelly, and K. Scheinberg, "Stochastic optimization using a trust-region method and random models," *Math. Program.*, vol. 169, no. 2, pp. 447–487, 2018. doi: 10.1007/s10107-017-1141-8.

[9] F. E. Curtis, D. P. Robinson, and M. Samadi, "A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization," *Math. Program.*, vol. 162, no. 1, pp. 1–32, 2017.

[10] F. E. Curtis and K. Scheinberg, "Optimization methods for supervised machine learning: from linear models to deep learning," in *INFORMS Tutorials in Operations Research*, R. Batta and J. Peng, Eds. Catonsville, MD: Institute for Operations Research and the Management Sciences *(INFORMS),* 2017, ch. 5, pp. 89–114.

[11] F. E. Curtis, K. Scheinberg, and R. Shi, "A stochastic trust region algorithm based on careful step normalization," *INFORMS J. Optim.*, vol. 1, no. 3, pp. 200–220, 2019. doi: 10.1287/ijoo.2018.0010.

[12] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Proc. 27th Int. Conf. Neural Information Processing Systems (NIPS)*, 2014, pp. 1646–1654.

[13] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.

[14] M. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *SIAM J. Sci. Comput.*, vol. 34, no. 3, pp. 1380–1405, 2012. doi: 10.1137/110830629.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[16] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang, "Complexity and global rates of trust-region methods based on probabilistic models," *IMA J. Numer. Anal.*, vol. 38, no. 3, pp. 1579–1597, Aug. 2017. doi: 10.1093/imanum/drx043.

[17] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. 26th Int. Conf. Neural Information Processing Systems (NIPS'13)*, 2013, pp. 315–323.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learning Representations (ICLR 2015)*, San Diego, May 7–9, 2015.

[19] J. Larson, M. Menickelly, and S. M. Wild, "Derivative-free optimization methods," *Acta Numerica*, vol. 28, pp. 287–404, Apr. 2019. doi: 10.1017/S0962492919000060.

[20] M. Mahsereci and P. Hennig, "Probabilistic line searches for stochastic optimization," *J. Mach. Learn. Res.*, vol. 18, no. 119, pp. 1–59, 2017.

[21] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, 2nd ed. Cambridge, MA: MIT Press, 2018.

[22] Y. Nesterov and B. T. Polyak, "Cubic regularization of newton method and its global performance," *Math. Program.*, vol. 108, no. 1, pp. 177–205, 2006. doi: 10.1007/s10107-006-0706-8.

[23] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takácˇ, "SARAH: A novel method for machine learning problems using stochastic recursive gradient," in *Proc. 34th Int. Conf. Machine Learning (ICML'17)*, 2017, pp. 2613–2621.

[24] C. Paquette and K. Scheinberg, "A stochastic line search method with expected complexity analysis," *SIAM J. Optim.*, vol. 30, no. 1, pp. 349–376, 2020. doi: 10.1137/18M1216250.

[25] R. Pasupathy, P. Glynn, S. Ghosh, and F. S. Hashemi, "On sampling rates in simulation-based recursions," *SIAM J. Optim.*, vol. 28, no. 1, pp. 45–73, 2018. doi: 10.1137/140951679.

[26] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 1951. doi: 10.1214/aoms/1177729586.

[27] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Math. Program.*, vol. 162, no. 1–2, pp. 83–112, 2017. doi: 10.1007/s10107-016-1030-6.

[28] N. Tripuraneni, M. Stern, C. Jin, J. Regier, and M. I. Jordan, "Stochastic cubic regularization for fast nonconvex optimization," in *Proc. 32nd Int. Conf. Neural Information Processing Systems (NeurIPS'18)*, 2018, pp. 2904–2913.

[29] S. Vaswani, A. Mishkin, I. H. Laradji, M. W. Schmidt, G. Gidel, and S. Lacoste-Julien, "Painless stochastic gradient: Interpolation, line-search, and convergence rates," in *Proc. 33rd Int. Conf. Neural Information Processing Systems (NeurIPS'19)*, 2019, pp. 3732–3745.

[30] P. Xu, F. Roosta, and M. W. Mahoney, "Newton-type methods for non-convex optimization under inexact Hessian information," *Math. Program.*, pp. 1–36, 2019. doi: 10.1007/s10107-019-01405-z.