SIAM J. SCI. COMPUT.
Vol. 34, No. 3, pp. A1380–A1405


# HYBRID DETERMINISTIC-STOCHASTIC METHODS FOR DATA FITTING*

MICHAEL P. FRIEDLANDER† AND MARK SCHMIDT‡

**Abstract.** Many structured data-fitting applications require the solution of an optimization problem involving a sum over a potentially large number of measurements. Incremental gradient algorithms offer inexpensive iterations by sampling a subset of the terms in the sum; these methods can make great progress initially, but often slow as they approach a solution. In contrast, full-gradient methods achieve steady convergence at the expense of evaluating the full objective and gradient on each iteration. We explore hybrid methods that exhibit the benefits of both approaches. Rate-of-convergence analysis shows that by controlling the sample size in an incremental-gradient algorithm, it is possible to maintain the steady convergence rates of full-gradient methods. We detail a practical quasi-Newton implementation based on this approach. Numerical experiments illustrate its potential benefits.

**Key words.** optimization, data fitting, incremental gradient, gradient descent

**AMS subject classifications.** 47N10, 90C06, 90C25, 94A20

**DOI.** 10.1137/110830629

## 1. Introduction.

Data-fitting applications are often typified by optimization problems of the form

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) := \frac{1}{M}\sum_{i=1}^{M} f_i(x), \tag{1.1}$$

where each function $f_i$ corresponds to a single observation (or measurement) and models the misfit for a given choice of parameters $x$. The aim is to choose parameters that minimize the misfit (or loss) across all measurements. The canonical example is least squares, and in that case,

$$f_i(x) = \tfrac{1}{2}(a_i^T x - b_i)^2 \quad \text{for all} \quad i = 1,\dots,M.$$

This misfit model corresponds to a linear model with Gaussian errors on the measurements $b$. For applications where the measurements $b$ are binary, a more appropriate model is logistic regression, described by the choice

$$f_i(x) = \log(1+\exp[-b_i a_i^T x]) \quad \text{for all} \quad i = 1,\dots,M.$$

These are both special cases of the more general maximum-likelihood problem. The maximum-likelihood approach gives rise to separable problems like (1.1) whenever the measurements $(a_i, b_i)$ are assumed to be independent and identically distributed.

*Submitted to the journal's Methods and Algorithms for Scientific Computing section April 13, 2011; accepted for publication (in revised form) April 6, 2012; published electronically May 29, 2012.
http://www.siam.org/journals/sisc/34-3/83062.html

†Department of Computer Science, University of British Columbia, Vancouver V6T 1Z4, B.C., Canada (mpf@cs.ubc.ca). The work of this author was supported by NSERC Discovery Grant 312104.

‡INRIA–SIERRA team, Laboratoire d'Informatique de l'Ecole Normale Superieure (INRIA/ENS/CNRS UMR 8548), 23 avenue d'Italie, 75214 Paris CEDEX 13, France (mark.schmidt@inria.fr). The work of this author was supported by the CRD DNOISE II NSERC grant, and the SIERRA grant from the European Research Council (SIERRA-ERC-239993).

If the number of measurements $M$ is very large, or if the individual $f_i$ are complicated functions (e.g., each $f_i$ evaluation may require the solution of a partial differential equation), then evaluating $f(x)$ and $\nabla f(x)$ can be computationally expensive. However, there is often a large amount of uniformity in the measurements, which means that a full evaluation of $f(x)$ and $\nabla f(x)$ may be unnecessary to make progress in solving (1.1). This motivates *incremental-gradient* methods, in which each iteration only evaluates the gradient with respect to a single $f_i$ [3, section 3.2].

Incremental-gradient methods enjoy an iteration cost that is $M$ times faster than full-gradient methods because the iterations are independent of $M$. Thus, in the time it takes to make one full-gradient iteration, the incremental-gradient method can achieve $M$ iterations, which often results in rapid initial progress. However, the number of iterations needed to reach the same level of accuracy may be much higher. Indeed, because of their faster convergence rate, full-gradient methods must eventually dominate incremental-gradient methods.

Our aim is to develop a method that exhibits the benefits of these two extremes. The approach is based on starting with iterations that resemble an incremental-gradient approach and use relatively few measurements to approximate the gradient; as the iterations proceed, the algorithm gradually increases the number of measurements. This preserves the rapid initial progress of incremental-gradient methods without sacrificing the convergence rate of full-gradient methods.

**1.1. Gradient descent with error.** In the most basic version of the algorithm that we consider, each iterate is computed via the update

$$x_{k+1} = x_k - \alpha_k g_k \tag{1.2}$$

for some step size $\alpha_k$; the search direction

$$g_k := \nabla f(x_k) + e_k \tag{1.3}$$

is an approximation of the gradient, and $e_k$ is the residual in its computation. This framework is the point of departure in the analysis of sections 2–3. Evidently, the full-gradient method (i.e., steepest descent) corresponds to the case where $e_k \equiv 0$, which means that the gradient is exactly computed at each iteration.

The stochastic approximation method deals with the case where the residual $e_k$ is a random variable; see [3]. In this context we typically assume that $\mathbb{E}[e_k] = 0$ and $\mathbb{E}[\|e_k\|^2] \leq B$ for some constant $B$, which are sufficient to guarantee that the iterates converge in a probabilistic sense (for a suitable choice of decreasing step sizes). Incremental-gradient methods are a special case of stochastic approximation. Here, rather than computing the full gradient $\nabla f(x_k)$ on each iteration, a function $f_i$ is randomly selected among $i \in \{1, \ldots, M\}$, and the gradient estimate is constructed as $g_k = \nabla f_i(x_k)$.

In this work we consider scenarios in which the gradient residual can be controlled on each iteration by specifying a per-iteration bound $B_k$ on the norm of the residual. In particular, we characterize convergence of the gradient-with-error algorithm (1.2)–(1.3) under two different conditions:

$$\text{deterministic:} \quad \|e_k\|^2 \leq B_k; \qquad \text{or} \qquad \text{stochastic:} \quad \mathbb{E}[\|e_k\|^2] \leq B_k. \tag{1.4}$$

Our analysis applies whether the noise is deterministic or stochastic, and we do not assume that the noise has zero mean. In the context of problem (1.1), the error in the gradient is a result of using a *sample* of the $f_i$ functions (sometimes referred to

as a *batch*). In section 3 we show how the *sample size* (or *batch size*) influences the bound $B_k$.

We also consider in section 4 the case in which the approximate gradient $g_k$ is scaled to account for curvature information in $f$. In this case, the iteration update is

$$x_{k+1} = x_k - \alpha_k d_k, \tag{1.5}$$

where $d_k$ solves the system

$$H_k d = -g_k, \tag{1.6}$$

and $H_k$ is a positive-definite approximation (e.g., a quasi-Newton Hessian) to $\nabla^2 f$.

**1.2. Assumptions and notation.** We make the blanket assumptions throughout that a minimizer $x_*$ of $f$ always exists, that the functions $f_i : \mathbb{R}^n \to \mathbb{R}$ are continuously differentiable, and that the overall gradient of $f$ is uniformly Lipschitz continuous, i.e., for some positive $L$,

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\| \quad \text{for all} \quad x, y \in \mathbb{R}^n. \tag{1.7a}$$

We also assume that $f$ is strongly convex with (positive) parameter $\mu$:

$$f(y) \ge f(x) + (y - x)^T \nabla f(x) + \tfrac{1}{2}\mu\|y - x\|^2 \quad \text{for all} \quad x, y \in \mathbb{R}^n. \tag{1.7b}$$

If $f$ is twice-continuously differentiable, then these assumptions are equivalent to the condition that the eigenvalues of the Hessian are uniformly bounded above and below:

$$\mu I \preceq \nabla^2 f(x) \preceq LI.$$

The ratio $L/\mu \ge 1$ is known as the *condition* number of $f$ [26, section 2.1.3].

We make the assumption—standard in the stochastic optimization literature [3, section 4.2]—that

$$\|\nabla f_i(x)\|^2 \le \beta_1 + \beta_2\|\nabla f(x)\|^2 \quad \text{for all } x \text{ and } i = 1, \dots, M, \tag{1.8}$$

for some constants $\beta_1 \ge 0$ and $\beta_2 \ge 1$. This implies that $\|e_k\|$ is bounded as a function of the true gradient of the objective.

We describe two versions of linear convergence of function values. The first is denoted as *weak* linear convergence and is characterized by a bounding sequence on the function value at every iteration $k$:

$$f(x_k) - f(x_*) = \mathcal{O}(\sigma^k) \quad \text{for some} \quad \sigma < 1. \tag{1.9}$$

This is a nonasymptotic version of R-linear convergence. (See, for example, [28, section A.2].) The second is denoted as *strong* linear convergence and characterizes the decrease of the function value at every iteration $k$:

$$f(x_{k+1}) - f(x_*) \le \sigma[f(x_k) - f(x_*)] \quad \text{for some} \quad \sigma < 1. \tag{1.10}$$

We emphasize that this inequality applies to all iterations of the algorithm and is a nonasymptotic version of Q-linear convergence. Note that (1.10) implies (1.9), though the converse is not implied.

All of the convergence results that we analyze are described only in terms of convergence of the objective function values and not the iterates $x_k$. This is sufficient,

however, because the strong convexity assumption allows us to directly deduce a convergence rate of $x_k$ to $x_*$ via the corresponding function values. In particular, strong convexity of $f$ implies

$$\frac{\mu}{2}\|x_k - x_*\|^2 \leq f(x_k) - f(x_*). \tag{1.11}$$

Thus, the rate at which the squared error $\|x_k - x_*\|^2$ converges is at least as fast as the rate at which $f(x_k)$ converges to the optimal value $f(x_*)$.

If the matrix $H_k$ in the iterations (1.5)–(1.6) is uniformly positive definite and uniformly bounded in norm (as can be enforced in practice), then assumptions (1.7a)–(1.7b) can be replaced by the following conditions: there exist positive constants $L'$ and $\mu'$ such that for all $x, y \in \mathbb{R}^n$ and for all $k = 0, 1, \ldots,$

$$\|\nabla f(x) - \nabla f(y)\|_{H_k^{-1}} \leq L'\|x - y\|_{H_k}, \tag{1.12a}$$

$$f(y) \geq f(x) + (y - x)^T \nabla f(x) + \tfrac{1}{2}\mu'\|x - y\|_{H_k}^2, \tag{1.12b}$$

where the quadratic norm $\|x\|_{H_k} = \sqrt{x^T H_k x}$ and its dual $\|x\|_{H_k^{-1}}$ are used instead of the Euclidean norm. It can then be verified that all of the results in Sections 2–3 apply to the Newton-like algorithm (1.5)–(1.6), where the parameters $L$ and $\mu$ in those results are replaced by the parameters $L'$ and $\mu'$ found in (1.12). The benefit of this approach is that a judicious choice of the scaling $H_k$ can lead to a scaled condition number $L'/\mu'$ that can be smaller than the condition number $L/\mu$ of the unscaled objective $f$, effectively improving on the error constants found in the convergence results.

**1.3. Contributions.** This paper is divided into six components.

*Weak linear convergence with generic bounds* (section 2.1). We analyze the convergence rate under a generic sequence $\{B_k\}$. Our results imply that for any (sub)linearly decreasing sequence $\{B_k\}$, the algorithm has a weak (sub)linear convergence rate. In the expected-error version of (1.4), the convergence rate is described in terms of the expected function value.

*Strong linear convergence with particular bounds* (section 2.2). We describe a particular construction of the sequence $\{B_k\}$ that ensures that the algorithm has a strong linear convergence rate. The rate achieved under this sequence can be arbitrarily close to the rate of the standard gradient method without error, without requiring an exact gradient calculation on any iteration.

*Sublinear convergence without strong convexity* (section 2.3). Without the strong convexity assumption on $f$, the convergence rate for the deterministic gradient method is sublinear. We show that a summable sequence $\{B_k\}$ is sufficient to maintain the same sublinear rate.

*Application to sample-average gradients* (section 3). For data-fitting problems of the form (1.1), we show that a growing sample-size strategy can be used as a mechanism for controlling the error in the estimated gradient and achieving a linear rate. In effect, choosing the sample size allows us to control the error, as in (1.4). By growing the sample size sufficiently fast, we implicitly set the rate at which $B_k \to 0$, and hence set the overall rate of the algorithm.

*A practical quasi-Newton implementation* (section 4). We describe a practical implementation of the ideas based on a limited-memory quasi-Newton approximation and a heuristic line search.

*Numerical results* (section 5). We evaluate the implementation on a variety of data-fitting applications (comparing it to incremental-gradient methods and a deterministic quasi-Newton method).

**1.4. Related work.** Our approach is based on bridging the gap between two ends of a spectrum, where incremental gradient methods are at the end of "cheap iterations with slow convergence," and full-gradient methods are "expensive iterations with fast convergence." In particular, incremental-gradient methods achieve an expected sublinear convergence rate on the expected value of $f(x_k)$, i.e.,

$$\mathbb{E}[f(x_k) - f(x_*)] = \mathcal{O}(1/k),$$

where the iterations are described by (1.2) for $\alpha_k = \mathcal{O}(1/k)$ [25, section 2.1]. In fact, among all first-order methods, this is the best possible dependency on $k$ given only a first-order stochastic oracle; thus a linear rate is not possible [24, section 14.1].

In contrast, consider the basic gradient-descent iteration (1.2) with a fixed step size $\alpha_k = 1/L$ and $g_k = \nabla f(x_k)$. It is well known that this algorithm has a strong linear convergence rate and satisfies the per-iteration decrease in (1.10) with $\sigma = 1 - \mu/L$; see [17, section 8.6].

Some authors have analyzed incremental-gradient methods with a constant step size [23]. Although this strategy does not converge to the optimal solution, it does converge at a linear rate to a neighborhood of the solution (where the size of the neighborhood increases with the step size).

In the context of incremental-gradient methods, several other hybrid methods have been proposed that achieve a linear convergence rate. The works of Bertsekas [2] and Blatt, Hero, and Gauchman [6] are the closest in spirit to our proposed approach. However, the convergence rates for these methods treat full passes through the data as iterations, similar to the full-gradient method. Further, there are numerical difficulties in evaluating certain sequences associated with the method of Bertsekas, while the method of Blatt, Hero, and Gauchman may require an excessive amount of memory.

This is not the first work to examine a growing sample-size strategy. This type of strategy appears to be a "folk" algorithm used by practitioners in several application domains, and it is explicitly mentioned in an informal context by some authors such as Bertsekas and Tsitsiklis [3, page 113], who refer to growing sample-size techniques as "batching" strategies. This is the first work, that we are aware of, that presents a theoretical analysis of the technique and that proposes a practical large-scale quasi-Newton implementation along with an experimental evaluation.

Gradient descent with a decreasing sequence of errors in the gradient measurement was previously analyzed by Luo and Tseng [18], and they present analogous weak and strong linear convergence results depending on the sequence of bounds on the noise. Our analysis extends this previous work in several ways:

- The weak linear convergence rate described in [18] requires a per-iteration strict decrease in the objective. In contrast, our analysis in section 2.1 does not require this assumption and allows for the (realistic) possibility that the noisy gradient can lead to an increase in the objective function on some iterations.
- The strong linear convergence rate shown in [18] holds only asymptotically, while the construction we give (see section 2.2) leads to a nonasymptotic rate of the form (1.10) that applies to all iterations of the algorithm.

- Luo and Tseng [18] consider deterministic errors in the gradient measurement that can be bounded in an absolute sense; we also consider the more general scenario where the error is stochastic and can only be bounded in expectation.

**1.5. Reproducible research.** Following the discipline of reproducible research, the source code and data files required to reproduce the experimental results of this paper can be downloaded from http://www.cs.ubc.ca/labs/scl/FriedlanderSchmidt2011.

**2. Convergence analysis.** Our convergence analysis first considers a basic first-order method with the constant step size $\alpha_k = 1/L$. The following intermediate result establishes an upper bound on the objective value at each iteration in terms of the residual in the computed gradient.

LEMMA 2.1. *At each iteration $k$ of algorithm* (1.2)*, with $\alpha_k \equiv 1/L$,*

$$f(x_{k+1}) - f(x_*) \le (1 - \mu/L)[f(x_k) - f(x_*)] + \frac{1}{2L}\|e_k\|^2. \tag{2.1}$$

*Proof.* It follows from assumptions (1.7) that the following inequalities hold:

$$f(y) \le f(x) + (y - x)^T \nabla f(x) + \frac{L}{2}\|y - x\|^2, \tag{2.2a}$$

$$f(y) \ge f(x) + (y - x)^T \nabla f(x) + \frac{\mu}{2}\|y - x\|^2. \tag{2.2b}$$

Use $x = x_k$ and $y = x_k - (1/L)g_k$ in (2.2a) and simplify to obtain

$$f(x_k - (1/L)g(x_k)) \le f(x_k) - \frac{1}{L}g(x_k)^T \nabla f(x_k) + \frac{1}{2L}\|g(x_k)\|^2.$$

Next, use the definitions of $x_{k+1}$ and $g_k$ (cf. (1.2)–(1.3)) in this expression to obtain

$$\begin{aligned} f(x_{k+1}) &\le f(x_k) - \frac{1}{L}(\nabla f(x_k) + e_k)^T \nabla f(x_k) + \frac{1}{2L}\|\nabla f(x_k) + e_k\|^2 \\ &= f(x_k) - \frac{1}{L}\|\nabla f(x_k)\|^2 - \frac{1}{L}\nabla f(x_k)^T e_k \\ &\qquad + \frac{1}{2L}\|\nabla f(x_k)\|^2 + \frac{1}{L}\nabla f(x_k)^T e_k + \frac{1}{2L}\|e_k\|^2 \\ &= f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2 + \frac{1}{2L}\|e_k\|^2. \end{aligned} \tag{2.3}$$

We now use (2.2b) to derive a lower bound on the norm of $\nabla f(x_k)$ in terms of the optimality of $f(x_k)$. Do this by minimizing both sides of (2.2b) with respect to $y$: by definition, the minimum of the left-hand side is achieved by $y = x_*$; the minimizer of the right-hand side is given by $y = x - (1/\mu)\nabla f(x)$. Thus,

$$f(x_*) \ge f(x) - \frac{1}{\mu}\nabla f(x)^T \nabla f(x) + \frac{1}{2\mu}\nabla f(x)^T \nabla f(x) = f(x) - \frac{1}{2\mu}\|\nabla f(x)\|^2$$

for any $x$. Rearranging and specializing to the case where $x = x_k$,

$$\|\nabla f(x_k)\|^2 \ge 2\mu[f(x_k) - f(x_*)]. \tag{2.4}$$

Subtract $f(x_*)$ from both sides of (2.3) and use (2.4) to get

$$\begin{aligned} f(x_{k+1}) - f(x_*) &\le f(x_k) - f(x_*) - \frac{\mu}{L}[f(x_k) - f(x_*)] + \frac{1}{2L}\|e_k\|^2 \\ &= (1 - \mu/L)[f(x_k) - f(x_*)] + \frac{1}{2L}\|e_k\|^2, \end{aligned}$$

which gives the required result. ∎

As an aside, note that (2.3) shows that the objective decreases monotonically, i.e., $f(x_{k+1}) < f(x_k)$ if $\|e_k\| < \|\nabla f(x_k)\|$. In general, however, we do not require this condition.

**2.1. Weak linear convergence.** In this section we show that if $\{B_k\}$ is any (sub)linearly convergent sequence, then algorithm (1.2) has a (sub)linear convergence rate. This result reflects that the convergence rate of the approximate gradient algorithm is not better than the rate at which the noise goes to zero, and of course is also not better than the rate of the noiseless algorithm.

THEOREM 2.2 (weak convergence rate under absolute error bounds). *Suppose that* $\|e_k\|^2 \le B_k$, *where*

$$\lim_{k\to\infty} B_{k+1}/B_k \le 1. \tag{2.5}$$

*Then at each iteration of algorithm* (1.2) *with* $\alpha_k \equiv 1/L$, *for any* $\epsilon > 0$ *we have*

$$f(x_k) - f(x_*) \le (1-\mu/L)^k[f(x_0) - f(x_*)] + \mathcal{O}(C_k),$$

*where* $C_k = \max\{B_k, (1-\mu/L+\epsilon)^k\}$.

*Proof.* Let $\rho := 1-\mu/L$. Because $\|e_k\|^2 \le B_k$, Lemma 2.1 implies

$$f(x_{k+1}) - f(x_*) \le [f(x_k) - f(x_*)]\rho + \frac{1}{2L}B_k.$$

Applying this recursively,

$$f(x_k) - f(x_*) = [f(x_0) - f(x_*)]\rho^k + \mathcal{O}(\mu_k),$$

where $\mu_k := \sum_{i=0}^{k-1} \rho^{k-i-1}B_i$. Observe that $\mu_{k+1} = B_k + \rho\mu_k$.

We now show that $\mu_k \le \xi C_k$ for all $k$ for some $\xi > 0$. It follows from (2.5) and the definition of $C_k$ that $\lim_{k\to\infty} C_{k+1}/C_k \ge \rho+\epsilon$ and thus that there exists some $N$ such that $C_{k+1}/C_k - \rho \ge \epsilon/2$ for all $k \ge N$. We now choose $\xi$ such that

$$\mu_k \le \xi C_k \text{ for all } k \le N, \quad \text{and} \quad \xi\epsilon/2 \ge 1.$$

This is always possible because the $\mu_k$ are finite and $\epsilon > 0$. We now show by induction that $\mu_k \le \xi C_k$ for all $k$. This trivially holds for all $k \le N$ by the definition of $\xi$. Assuming this holds for some arbitrary $k \ge N$, we have

$$\begin{aligned}\mu_{k+1} = B_k + \rho\mu_k &\le C_k + \rho\xi C_k \\ &\le \xi(\epsilon/2)C_k + \rho\xi C_k \le \xi(C_{k+1}/C_k - \rho)C_k + \rho\xi C_k = \xi C_{k+1}.\end{aligned}$$

Thus, $\mu_k = \mathcal{O}(C_k)$ for all $k$, as required. ∎

An implication of this result is that the algorithm has a linear convergence rate if $B_k$ converges linearly to zero. For example, if $B_k = \mathcal{O}(\gamma^k)$ with $\gamma < 1$, then

$$f(x_k) - f(x_*) = \mathcal{O}(\sigma^k),$$

where $\sigma = \max\{\gamma, (1-\mu/L+\epsilon)\}$ for any positive $\epsilon < \mu/L$.

Theorem 2.2 also yields a convergence rate in scenarios where the high cost of computing an accurate gradient might make it appealing to allow the error in the gradient measurement to decrease sublinearly. For example, if $B_k = \mathcal{O}(1/k^2)$, then $f(x_k) - f(x_*) = \mathcal{O}(1/k^2)$, which is the rate achieved by Nesterov's optimal method for

general smooth convex (but not necessarily strongly convex) functions in the noiseless setting [26, section 2.1]. Theorem 2.2 also allows for the possibility that the bound $B_k$ does not converge to zero (necessarily implying the limit of $B_{k+1}/B_k$ is one). In this case, the result simply states that the distance to optimality is eventually bounded by a constant times $B_k$.

The above analysis allows for the possibility that the approximate gradient is computed by a stochastic algorithm where the error made by the algorithm can be bounded in an absolute sense. We now consider the more general case where the error can only be bounded in expectation. The following result is the counterpart to Theorem 2.2, where we instead have a bound on the expected value of $\|e_k\|^2$.

THEOREM 2.3 (weak expected convergence rate under expected error bounds). *Suppose that* $\mathbb{E}[\|e_k\|^2] \leq B_k$, *where*

$$\lim_{k\to\infty} B_{k+1}/B_k \leq 1.$$

*Then at each iteration of algorithm* (1.2) *with* $\alpha_k \equiv 1/L$, *for any* $\epsilon > 0$ *we have*

$$\mathbb{E}[f(x_k) - f(x_*)] \leq (1-\mu/L)^k[f(x_0) - f(x_*)] + \mathcal{O}(C_k),$$

*where* $C_k = \max\{B_k, (1-\mu/L+\epsilon)^k\}$.

*Proof.* We use Lemma 2.1 and take expectations of (2.1) to obtain

$$\mathbb{E}[f(x_{k+1}) - f(x_*)] \leq (1-\mu/L)\mathbb{E}[f(x_k) - f(x_*)] + \frac{1}{2L}\mathbb{E}[\|e_k\|^2].$$

Proceeding as in the proof of Theorem 2.2, we obtain a similar result, but based on the expected value of the objective. ∎

**2.2. Strong linear convergence.** We now describe a particular construction of the sequence $\{B_k\}$ that allows us to achieve a linear decrease of the function values that applies to every iteration $k$. This strong guarantee, however, comes at the price of requiring bounds on three quantities that are unknowable for general problems: the strong convexity constant $\mu$, the gradient's Lipschitz constant $L$, and a nontrivial lower bound on the current iterate's distance to optimality $f(x_k) - f(x_*)$.

In particular, we consider any sequence $\{B_k\}$ that satisfies

$$0 \leq B_k \leq 2L(\mu/L - \rho)\pi_k, \quad k = 0, 1, \ldots, \tag{2.6}$$

where $\rho \leq \mu/L$ is a positive constant, which controls the convergence rate, and $\pi_k$ is a nonnegative lower bound on the distance to optimality, i.e.,

$$0 \leq \pi_k \leq f(x_k) - f(x_*). \tag{2.7}$$

THEOREM 2.4 (strong linear rate under absolute error bounds). *Suppose that* $\|e_k\|^2 \leq B_k$ *where* $B_k$ *is given by* (2.6). *Then at each iteration of algorithm* (1.2) *with* $\alpha_k \equiv 1/L$,

$$f(x_{k+1}) - f(x_*) \leq (1-\rho)[f(x_k) - f(x_*)].$$

*Proof.* Because $\|e_k\|^2 \leq B_k$, Lemma 2.1 and (2.7) imply that

$$\begin{aligned} f(x_{k+1}) - f(x_*) &\leq (1-\mu/L)[f(x_k) - f(x_*)] + \frac{1}{2L}B_k \\ &\leq (1-\mu/L)[f(x_k) - f(x_*)] + (\mu/L - \rho)\pi_k \\ &\leq (1-\mu/L)[f(x_k) - f(x_*)] + (\mu/L - \rho)[f(x_k) - f(x_*)] \\ &= (1-\rho)[f(x_k) - f(x_*)], \end{aligned}$$

as required. ∎

As we did with Theorem 2.3, we consider the case where the approximate gradient is computed by a stochastic algorithm, and the error can only be bounded in expectation. Provided we now have a lower bound $\pi_k$ on the expected suboptimality, i.e.,

$$0 \le \pi_k \le \mathbb{E}[f(x_k) - f(x_*)],$$

it is possible to show an expected linear convergence rate that parallels Theorem 2.4. The proof follows that of Theorem 2.4, where we instead begin by taking the expectation of both sides of (2.1).

THEOREM 2.5 (strong expected linear rate under expected error bounds). *Suppose that* $\mathbb{E}[\|e_k\|^2] \le B_k$, *where* $B_k$ *is given by* (2.6). *Then at each iteration of algorithm* (1.2) *with* $\alpha_k \equiv 1/L$,

$$\mathbb{E}[f(x_{k+1}) - f(x_*)] \le (1-\rho)\mathbb{E}[f(x_k) - f(x_*)].$$

In both scenarios, we obtain the fastest convergence rate in the extreme case where $\rho = \mu/L$. From (2.6), this means that $B_k$ must be zero, i.e., the gradient is exact, and we obtain the classic strong linear convergence result with error constant $\sigma = 1 - \mu/L$ stated in section 1.4. However, if we take any positive $\rho$ less than $\mu/L$, then we obtain a slower linear convergence rate, but (2.6) allows $B_k$ to be nonzero (as long as $\pi_k > 0$).

The bound (2.6) on the error depends on both the conditioning of the problem (as determined by the parameter $\mu$ and $L$), and the lower bound on the distance to the optimal function value $\pi_k$. This seems intuitive. For example, we can allow a larger error in the gradient calculation the further the current iterate is from the optimal function value, but a more accurate calculation is needed to maintain the strong linear convergence rate as the iterates approach the solution. Similarly, if the problem is well conditioned so that the ratio $\mu/L$ is close to 1, a larger error in the gradient calculation is permitted, but for ill-conditioned problems where $\mu/L$ is very small, we require a more accurate gradient calculation.

Note that the analysis of this section holds even if the bounds $\mu$, $L$, and $\pi_k$ are not the tightest possible. Unsurprisingly, we obtain the fastest convergence rate when $\mu$ is as large and $L$ is as small as possible, while the largest error in the gradient calculation is allowed if $\pi_k$ is similarly as large as possible. Note that with $\pi_k = 0$, which is a trivial bound that is valid by definition, we require an exact gradient.

Although it is difficult in general to obtain a bound $\pi_k$ that satisfies (2.7), there are at least two possible heuristics that could be used in practice to approximate this quantity. The first heuristic is possible if we have bounds on the Lipschitz and strong-convexity constants ($L$ and $\mu$), as well as the norm of the gradient at $x_k$. In that case, we use the stationarity of $x_*$, (1.7a), and (1.11) to deduce that

$$\frac{\mu}{2L^2}\|\nabla f(x_k)\|^2 \le \frac{\mu}{2}\|x_k - x_*\|^2 \le f(x_k) - f(x_*).$$

Although we typically do not have access to $\|\nabla f(x_k)\|$, the norm of its approximation $\|g_k\|$ is a reasonable proxy, which will improve as the magnitude of the gradient residual decreases.

The second heuristic is based on the assumption that the distance of consecutive iterates $x_k$ to the solution $x_*$ decreases monotonically. In that case,

$$\|x_k - x_{k+1}\|^2 = \|(x_k - x_*) - (x_{k+1} - x_*)\|^2 \le 4\|x_k - x_*\|^2.$$

Coupling this with (1.11) and premultiplying by $\mu/8$, we obtain the bound

$$\frac{\mu}{8}\|x_k - x_{k+1}\|^2 \leq f(x_k) - f(x_*).$$

This option may give a reasonable heuristic in the context of the increasing sample-size strategy even if the distance to the optimal solution does increase on some iterations, because as the sample size increases it becomes less likely that the distance will increase.

**2.3. Relaxing strong convexity.** If we remove the strong-convexity assumption, the deterministic gradient method has a sublinear convergence rate of $\mathcal{O}(1/k)$ [26, Section 2.1.5] while the stochastic gradient method under standard assumptions has a slower sublinear convergence rate of $\mathcal{O}(1/\sqrt{k})$ [24, Section 14.1]. Using an argument that does not rely on strong convexity, we show that the $\mathcal{O}(1/k)$ convergence rate of the deterministic gradient method is preserved for the average of the iterates if the residuals of the computed gradients are summable.

THEOREM 2.6 (sublinear rate under summable error bounds). *Suppose that* $\sum_{i=0}^{\infty}\|e_k\| < \infty$. *Then at each iteration of algorithm* (1.2) *with* $\alpha_k \equiv 1/L$,

$$f(\bar{x}_k) - f(x_*) = \mathcal{O}(1/k),$$

*where* $\bar{x}_k := (1/k)\sum_{i=1}^{k} x_i$.

*Proof.* As in (2.3), Lipschitz continuity of the gradient implies that

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2 + \frac{1}{2L}\|e_k\|^2.$$

We use convexity of $f$ to bound $f(x_k)$ and obtain

$$f(x_{k+1}) \leq f(x_*) + (x_k - x_*)^T\nabla f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2 + \frac{1}{2L}\|e_k\|^2.$$

Combine (1.2) and (1.3) to deduce that $-\nabla f(x_k) = e_k + L(x_{k+1} - x_k)$, and move $f(x_*)$ to the left-hand side to get, after simplifying,

$$\begin{aligned} f(x_{k+1}) - f(x_*) &= -(x_k - x_*)^T(e_k + L[x_{k+1} - x_k]) \\ &\quad - \frac{L}{2}\|x_{k+1} - x_k\|^2 - (x_{k+1} - x_k)^T e_k \\ &= \frac{L}{2}\|x_k - x_*\|^2 - (x_k - x_*)^T e_k - (x_{k+1} - x_k)^T e_k \\ &\quad - \frac{L}{2}\|x_k - x_*\|^2 - L(x_k - x_*)^T(x_{k+1} - x_k) - \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &\leq \frac{L}{2}\left[\|x_k - x_*\|^2 - \|x_{k+1} - x_*\|^2\right] + \|e_k\| \cdot \|x_{k+1} - x_*\|. \end{aligned}$$

Summing both sides up to iteration $k$, we get

$$\sum_{i=1}^{k}\left[f(x_i) - f(x_*)\right] \leq \frac{L}{2}\|x_0 - x_*\|^2 - \frac{L}{2}\|x_k - x_*\|^2 + \sum_{i=1}^{k}\|e_{i-1}\| \cdot \|x_i - x_*\|. \tag{2.8}$$

We will first use (2.8) to show that the sequence $\{\|x_k - x_*\|\}$ is bounded and then use it to obtain the final result.

Because the left-hand side of (2.8) is nonnegative,

$$\frac{L}{2}\|x_k - x_*\|^2 \le \frac{L}{2}\|x_0 - x_*\|^2 + \sum_{i=1}^{k} \|e_{i-1}\| \cdot \|x_i - x_*\|.$$

We now prove that the sequence $\{\|x_k - x_*\|\}$ is bounded by showing that the auxiliary sequence $\{d_k\}$, with $d_k := \max\{1, \|x_k - x_*\|\}$, is bounded. Because $d_k \ge 1$,

$$\begin{aligned} d_k \le d_k^2 &\le \max\left\{ 1,\, \|x_0 - x_*\|^2 + \frac{2}{L}\sum_{i=1}^{k} \|e_{i-1}\| \cdot \|x_i - x_*\| \right\} \\ &\le d_0^2 + \frac{2}{L}\sum_{i=1}^{k} \|e_{i-1}\| \cdot \|x_i - x_*\| \\ &\le d_0^2 + C_1 \sum_{i=1}^{k} \|e_{i-1}\| \cdot d_i, \end{aligned}$$

where $C_1 := 2/L$. Because the sequence $\{\|e_k\|\}$ is summable, it holds that $\|e_k\| \to 0$ and thus that there exists some $N$ large enough that $C_1\|e_{k-1}\| < 1$ for all $k \ge N$. Partition the sum at $N$:

$$d_k \le C_0 + C_1 \sum_{i=N}^{k} \|e_{i-1}\| \cdot d_i \quad \text{for all} \quad k > N,$$

where $C_0 := d_0^2 + C_1 \sum_{i=1}^{N-1} \|e_{i-1}\| \cdot d_i$. Rearrange terms to get

$$(1 - C_1\|e_{k-1}\|)d_k \le C_0 + C_1 \sum_{i=N}^{k-1} \|e_{i-1}\| \cdot d_i \quad \text{for all} \quad k > N,$$

and because $C_1\|e_{k-1}\| < 1$ for all $k > N$,

$$d_k \le \frac{1}{1 - C_1\|e_{k-1}\|}\left[ C_0 + C_1 \sum_{i=N}^{k-1} \|e_{i-1}\| d_i \right] \quad \text{for all} \quad k > N.$$

If we apply this bound recursively to $d_{k-1}$, we obtain

$$\begin{aligned} d_k &\le \frac{1}{1 - C_1\|e_{k-1}\|}\left[ C_0 + C_1 \sum_{i=N}^{k-2} \|e_{i-1}\| d_i + \frac{C_1\|e_{k-2}\|}{1 - C_1\|e_{k-2}\|}\left( C_0 + C_1 \sum_{i=N}^{k-2} \|e_{i-1}\| d_i \right) \right] \\ &= \frac{1}{(1 - C_1\|e_{k-1}\|)(1 - C_1\|e_{k-2}\|)}\left[ C_0 + C_1 \sum_{i=N}^{k-2} \|e_{i-1}\| d_i \right] \quad \text{for all} \quad k > N, \end{aligned}$$

and if we apply it recursively from $k$ down to $N$, we obtain

$$d_k \le \frac{2C_0}{\prod_{i=N}^{k}(1 - C_1\|e_{i-1}\|)} \le \frac{2C_0}{\prod_{i=N}^{\infty}(1 - C_1\|e_{i-1}\|)}. \tag{2.9}$$

To see that the right-hand side term is bounded, take logarithms of both sides to get

$$\log d_k \le \log(2C_0) - \sum_{i=N}^{\infty} \log(1 - C_1\|e_{i-1}\|).$$

We now use the limit-comparison test, which asserts that for nonnegative sequences $\{a_i\}$ and $\{b_i\}$, if $0 < \lim_{i\to\infty} a_i/b_i < \infty$, then $\sum_i b_i < \infty$ implies $\sum_i a_i < \infty$. We thus compare the sequence $\{-\log(1-C_1\|e_{i-1}\|)\}$ to the summable sequence $\{\|e_{i-1}\|\}$ using l'Hôpital's rule to get

$$\lim_{i\to\infty} \frac{-\log(1-C_1\|e_{i-1}\|)}{\|e_{i-1}\|} = \lim_{i\to\infty} \frac{\frac{C_1}{1-C_1\|e_{i-1}\|}}{1} = C_1.$$

Thus, the sequence $\{-\log(1-C_1\|e_{i-1}\|)\}$ is summable, which implies, via (2.9) and the definition of $d_k$, that the sequence $\{\|x_k - x_*\|\}$ is bounded.

We now use convexity of $f$ and (2.8) to bound the function value of the average iterate:

$$\begin{aligned} f\left(\frac{1}{k}\sum_{i=1}^{k} x_i\right) - f(x_*) &\le \frac{1}{k}\sum_{i=1}^{k}\left[f(x_i) - f(x_*)\right] \\ &\le \frac{L}{2k}\|x_0 - x_*\|^2 + \frac{1}{k}\sum_{i=1}^{k}\|e_{i-1}\|\cdot\|x_i - x_*\|. \end{aligned}$$

Because the sequence $\{\|e_k\|\}$ is summable (by assumption), and the sequence $\{\|x_k - x_*\|\}$ is bounded, this last inequality implies the conclusion of the theorem. ∎

Note that the convergence rate also holds for the iterate that achieves the lowest function value, but, unlike the deterministic case, this is not guaranteed to be the last iterate.

**3. Application to sample-average gradients.** Incremental-gradient methods for (1.1) are based on the iteration scheme (1.2) with the gradient approximation

$$g_k := \frac{1}{|\mathcal{B}_k|}\sum_{i\in\mathcal{B}_k} \nabla f_i(x_k),$$

where the set $\mathcal{B}_k \subseteq \{1,\dots,M\}$ represents a sample of the measurements that constitute the full data set. Typically, $\mathcal{B}_k$ contains a single element that is either chosen in a cyclic fashion or sampled at random, and, as discussed in section 1.4, the convergence rate of the method is sublinear. As the sample size increases, however, the error in the sampled gradient $g_k$ decreases, and so the sample size can be used to implicitly control the error in the gradient. We use the results of section 2 to develop an increasing sample-size strategy that improves on this sublinear rate.

Let $\mathcal{N}_k$ denote the complement of $\mathcal{B}_k$, so that $\mathcal{B}_k \cup \mathcal{N}_k = \{1,\dots,M\}$. Then the gradient residual, defined by (1.3), satisfies

$$e_k = \frac{M - |\mathcal{B}_k|}{M|\mathcal{B}_k|}\sum_{i\in\mathcal{B}_k} \nabla f_i(x_k) - \frac{1}{M}\sum_{i\in\mathcal{N}_k} \nabla f_i(x_k), \tag{3.1}$$

The first term is a reweighting of the gradient approximation, and the second term is the portion of $\nabla f(x_k)$ that is not sampled. Assumption (1.8) allows us to bound the norm of this residual in terms of the full gradient and thus leverage the results of section 2.

**3.1. Deterministic sampling.** It follows from assumption (1.8) that the gradient residual in (3.1) satisifies

$$\begin{aligned}
\|e_k\|^2 &= \left\| \left( \frac{M - |\mathcal{B}_k|}{M|\mathcal{B}_k|} \right) \sum_{i \in \mathcal{B}_k} \nabla f_i(x_k) - \frac{1}{M} \sum_{i \in \mathcal{N}_k} \nabla f_i(x_k) \right\|^2 \\
&\leq \left[ \left( \frac{M - |\mathcal{B}_k|}{M|\mathcal{B}_k|} \right) \Big\| \sum_{i \in \mathcal{B}_k} \nabla f_i(x_k) \Big\| + \frac{1}{M} \Big\| \sum_{i \in \mathcal{N}_k} \nabla f_i(x_k) \Big\| \right]^2 \\
&\leq \left[ \left( \frac{M - |\mathcal{B}_k|}{M|\mathcal{B}_k|} \right) \sum_{i \in \mathcal{B}_k} \|\nabla f_i(x_k)\| + \frac{1}{M} \sum_{i \in \mathcal{N}_k} \|\nabla f_i(x_k)\| \right]^2 \\
&\leq 4 \left( \frac{M - |\mathcal{B}_k|}{M} \right)^2 (\beta_1 + \beta_2 \|\nabla f(x_k)\|^2),
\end{aligned}$$

where the triangle-inequality is applied repeatedly and the resulting terms are simplified. Next, in the same way that (2.4) was derived, we use (2.2a) to derive the upper bound

$$\|\nabla f(x_k)\|^2 \leq 2L[f(x_k) - f(x_*)].$$

Thus the bound on $\|e_k\|^2$ can be expressed in terms of the sample-size ratio and the distance to optimality:

$$\|e_k\|^2 \leq 4 \left[ \frac{M - |\mathcal{B}_k|}{M} \right]^2 \big( \beta_1 + 2\beta_2 L[f(x_k) - f(x_*)] \big). \tag{3.2}$$

The following result parallels Theorem 2.2 and asserts that a linearly increasing sample size is sufficient to induce a weak linear convergence rate of the algorithm.

THEOREM 3.1 (weak linear rate with deterministic sampling). *Suppose that* (1.8) *holds and that the sample size* $|\mathcal{B}_k|$ *increases geometrically toward* $M$*, i.e.,*

$$\frac{M - |\mathcal{B}_k|}{M} = \mathcal{O}(\gamma^{k/2})$$

*for some* $\gamma < 1$*. Then at each iteration of algorithm* (1.2) *with* $\alpha_k \equiv 1/L$*, for any* $\epsilon > 0$ *we have*

$$f(x_k) - f(x_*) = [f(x_0) - f(x_*)]\mathcal{O}([1 - \mu/L + \epsilon]^k) + \mathcal{O}(\sigma^k),$$

*where* $\sigma = \max\{\gamma, 1 - \mu/L\} + \epsilon$.

*Proof.* Let $\rho_k = (M - |\mathcal{B}_k|)/M$. Using (3.2) and Lemma 2.1, we obtain the bound

$$\begin{aligned}
f(x_{k+1}) - f(x_*) &\leq (1 - \mu/L)[f(x_k) - f(x_*)] + \frac{2\rho_k}{L}(\beta_1 + 2\beta_2 L[f(x_k) - f(x_*)]) \\
&= (1 - \mu/L + 4\beta_2\rho_k)[f(x_k) - f(x_*)] + \frac{2\beta_1}{L}\rho_k \\
&= \omega_k[f(x_k) - f(x_*)] + \frac{2\beta_1}{L}\rho_k,
\end{aligned}$$

where $\omega_k := 1 - \mu/L + 4\beta_2\rho_k$. Applying this recursively, we obtain

$$f(x_k) - f(x_*) \leq [f(x_0) - f(x_*)][w_k]^k + \sum_{i=0}^{k-1} \mathcal{O}([\omega_k]^{k-i-1}\gamma^i).$$

We now take $\delta_k := \max\{\gamma, \omega_k\}$ and obtain

$$f(x_k) - f(x_*) \leq [f(x_0) - f(x_*)][w_k]^k + \mathcal{O}(k[\delta_k]^{k-1}).$$

Because $\rho_k \to 0$, it follows that $\omega_k \to 1 - \mu/L$ and $\delta_k \to \bar{\delta} := \max\{\gamma, 1 - \mu/L\}$. Thus $[\omega_k]^k = \mathcal{O}(1 - \mu/L + \epsilon)$ for any $\epsilon > 0$, which bounds the first term in the right-hand side above. To construct the bounding sequence $\{\xi\sigma^k\}$ for the second term in the right-hand side, Choose any $\sigma \in (\bar{\delta}, 1)$. Thus, $\sigma > \delta_k$ for all $k$ large enough and we subsequently choose $\xi \geq (k/\delta_k)(\delta_k/\sigma)^k$, which is possible because the maximum of the right-hand side exists. ∎

An interesting difference with Theorem 2.2, which considers a generic error in the gradient, is that the error in the objective function decreases at twice the rate that the sample size increases.

It is possible to get a strong linear rate of convergence by increasing the sample size in a more controlled way. Theorem 2.4 guides the choice of the sample size, and gives the following corollary. The proof follows by simply ensuring that the right-hand side of (3.2) is bounded as required by Theorem 2.4.

COROLLARY 3.2 (strong linear rate with deterministic sampling). *Suppose that* (1.8) *holds, and that the sample size* $|\mathcal{B}_k|$ *is increased so that at each iteration* $k = 0, 1, \ldots,$

$$4\left(\frac{M - |\mathcal{B}_k|}{M}\right)^2 \left(\beta_1 + 2\beta_2 L[f(x_k) - f(x_*)]\right) \leq L(\mu/L - \rho)[f(x_k) - f(x_*)]$$

*for some positive* $\rho \leq \mu/L$. *Then at each iteration of algorithm* (1.2) *with* $\alpha_k \equiv 1/L$,

$$f(x_{k+1}) - f(x_*) \leq (1 - \rho)[f(x_k) - f(x_*)].$$

We see that if the individual functions $f_i$ are very similar (so that $\beta_1$ and $\beta_2$ are small), then we can choose a fairly small sample size. In contrast, if the $f_i$ are very dissimilar, then we must use a larger sample size.

**3.2. Stochastic sampling.** Theorem 3.1 and Corollary 3.2 are based on the deterministic bound (3.2) on the gradient error and hold irrespective of the manner in which the elements of the samples $\mathcal{B}_k$ are chosen; e.g., the samples do not need to be chosen cyclically or sampled uniformly. We can obtain a strictly tighter bound in expectation, however, if we choose the sample by uniform sampling *without replacement*. In particular, by suitably modifying the derivation of a well-known result in statistical sampling, we can obtain a bound in terms of a quantity related to the sample variance of the gradients:

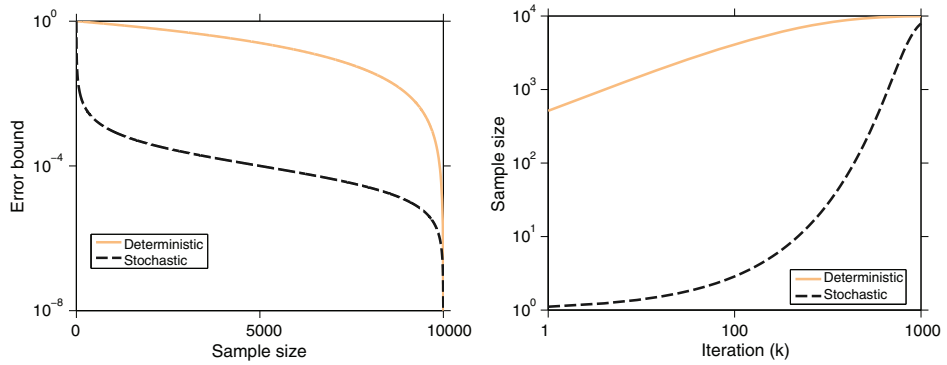
$$S_k = \frac{1}{M-1} \sum_{i=1}^{M} ||\nabla f_i(x_k) - \nabla f(x_k)||^2.$$

Using an argument similar to [16, section 2.7], uniform sampling *without replacement* yields

$$\mathbb{E}[\|e_k\|^2] = \left(\frac{M - |\mathcal{B}_k|}{M}\right) \frac{S_k}{|\mathcal{B}_k|}.$$

By using (1.8), we obtain the bound

$$\mathbb{E}[\|e_k\|^2] \leq \left[\frac{M - |\mathcal{B}_k|}{M} \cdot \frac{1}{|\mathcal{B}_k|}\right] \frac{M}{M-1}(\beta_1 + 2(\beta_2 - 1)L[f(x_k) - f(x_*)]).$$




(a) (b)

FIG. 3.1. (a) *Bounding factors in the error of the sampled gradient. The error bound for stochastic sampling is uniformly better than that for deterministic sampling.* (b) *Minimum sample-size schedule required to achieve a linear rate with error constant* 0.9. *In both figures,* $M = 10^4$.

An expected convergence result parallel to Theorem 3.1 can then be obtained with only minor changes to the proof.

THEOREM 3.3 (weak linear rate with stochastic sampling). *Suppose that* (1.8) *holds and that*

$$\frac{M - |\mathcal{B}_k|}{M} \cdot \frac{1}{|\mathcal{B}_k|} = \mathcal{O}(\gamma^k)$$

*for some* $\gamma < 1$. *Then at each iteration of algorithm* (1.2) *with* $\alpha_k \equiv 1/L$, *for any* $\epsilon > 0$ *we have*

$$\mathbb{E}[f(x_k) - f(x_*)] = [f(x_0) - f(x_*)]\mathcal{O}([1 - \mu/L + \epsilon]^k) + \mathcal{O}(\sigma^k),$$

*where* $\sigma = \max\{\gamma, 1 - \mu/L\} + \epsilon$.

Note that the right-hand side of the bound on $\mathbb{E}[\|e_k\|^2]$ is uniformly better than the bound shown in (3.2). Importantly, it initially decreases to zero at a faster rate as the size of the sample increases. Figure 3.1(a) illustrates the difference in the sample-size requirements between Theorems 3.1 and 3.3, i.e.,

$$\left[\frac{M - |\mathcal{B}_k|}{M}\right]^2 \quad \text{(deterministic)} \qquad \text{versus} \qquad \left[\frac{M - |\mathcal{B}_k|}{M} \cdot \frac{1}{|\mathcal{B}_k|}\right] \quad \text{(stochastic)},$$

as the sample size $|\mathcal{B}_k| \to M := 10^4$. Figure 3.1(b) illustrates the sample-size schedule needed to realize a linear convergence rate in the deterministic and stochastic cases.

**4. Practical implementation.** The analysis so far has focused on the approximate gradient descent iterations given by (1.2)–(1.3). In practice, it is useful to make two modifications to this basic algorithm. The first, described by (1.5)–(1.6), is based on scaling the search directions to account for curvature information in $f$. The second allows for a varying step size. For this section, we define the sampled objective function and its gradient by

$$\bar{f}_k(x) = \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} f_i(x) \quad \text{and} \quad g_k(x) = \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} \nabla f_i(x). \tag{4.1}$$

**4.1. Scaled direction.** Our implementation attempts to gather useful curvature information on the function $f$ by maintaining a quasi-Newton approximation to the Hessian $H_k$. The search directions $d_k$ are then taken as the solution of the system (1.6). The approximate Hessian is maintained via the recursive application of the update

$$H_{k+1} = U(H_k, y_k, s_k),$$

where $U$ represents an update formula on the $k$th iteration, while

$$s_k := x_{k+1} - x_k \quad \text{and} \quad y_k := g_k(x_{k+1}) - g_k(x_k)$$

measure the change in $x$ and the sampled gradient. In our experiments (see section 5) we use a limited-memory quasi-Newton update, which maintains a history of the previous $\ell = 10$ pairs $(s_k, y_k)$ and recursively applies the formula $H_{i+1} = U(H_i, s_i, y_i)$ for $i = k - \ell, \dots, k$. Nocedal and Wright [28, section 7.2] describe the recursive procedure for the limited-memory BFGS update that we use. Updates are skipped if necessary in order to ensure that the approximation $H_k$ remains positive definite and has a bounded condition number. To ensure that $d_k$ is well scaled, we use the Shanno and Phua [33] scaling of the initial Hessian approximation on each iteration as described by Nocedal and Wright [28, p. 178].

Another approach, not considered here, is to base the Hessian approximation on a set of sampled gradients [7].

**4.2. Varying stepsize.** A weakness of our convergence analysis is the requirement for a fixed steplength $\alpha_k \equiv L$, in part because the Lipschitz constant is not usually known, and in part because a dynamic steplength is typically more effective in practice. But a linesearch procedure that ensures a sufficient decrease condition in the true function $f$ runs contrary to a sampling scheme specifically designed to avoid expensive evaluations with $f$.

In our implementation we attempt to strike a balance between a rigorous linesearch and none at all by enforcing an Armijo-type descent condition on the sampled objective $\bar{f}_k$. In particular, we use a linesearch procedure to select a steplength $\alpha_k$ that satisfies

$$\bar{f}_k(x_k + \alpha d_k) < \bar{f}_k(x_k) + \eta \alpha g_k(x_k)^T d_k, \tag{4.2}$$

where $d_k$ is the current search direction and $\eta \in (0, 1)$. While in deterministic quasi-Newton methods we typically first test whether $\alpha = 1$ satisfies this condition, in our implementation we set our initial trial step length to $\alpha = |\mathcal{B}_{k-1}|/|\mathcal{B}_k|$.

In general, $\mathcal{B}_k$ represents only a fraction of all observations, and so the above procedure may not even yield a decrease in the true objective at each iteration. But because we steadily increase the sample size, the procedure just described eventually reduces to a conventional linesearch based on the true objective function $f$. Hence, the method may initially be nonmonotic but is guaranteed to eventually be monotonic. Coupled with the choice of search direction described in section 4.1, the overall algorithm reduces to a conventional linesearch method with a quasi-Newton Hessian approximation and inherits the global and local convergence guarantees of that method.

**5. Numerical experiments.** This section summarizes a series of numerical experiments in which we apply our incremental-gradient method with a growing sample size to a series of data-fitting applications. Table 5.1 summarizes the test problems.

The first four experiments are data-fitting applications of logistic regression of varying complexity: binary, multinomial, chain-structured conditional random fields

TABLE 5.1
*The test problems.*

| Problem | $M$ | $n$ | Description |
|---|---|---|---|
| binary logistic regression | 92,189 | 823,470 | spam identification (section 5.1) |
| multinormal logistic regression | 70,000 | 785 | digit identification (section 5.2) |
| chain-structured CRF | 8,936 | 1,643,004 | noun-phrase chunking (section 5.3) |
| general CRF | 50 | 4 | image denoising (section 5.4) |
| nonlinear least squares | 101 | 10,201 | seismic inversion (section 5.5) |

(CRFs) and general CRFs. These logistic-regression applications follow a standard pattern. We first model the probability of an outcome $b_i$ by some log-concave function $p(b_i \mid a_i, x)$, where $a_i$ is data; the goal is to choose the parameters $x$ so that the likelihood function

$$\mathcal{L}(x) = \prod_{i=1}^{M} p(b_i \mid a_i, x)$$

is maximized. We then approximate $x$ by minimizing the 2-norm-regularized negative log-likelihood function

$$f(x) = \sum_{i=1}^{M} f_i(x) + \tfrac{1}{2}\lambda\|x\|^2 \quad \text{with} \quad f_i(x) = -\log p(b_i \mid a_i, x) \tag{5.1}$$

for some positive regularization parameter $\lambda$. These objectives are all strongly convex (with $\mu \geq \lambda$) and satisfy our assumptions in section 1.2. For some functions $p$ the Lipschitz constant of $\nabla f$, or upper bounds on it, are available. We discuss the particular case of the binary model in section 5.1.

The last experiment is a more general application of nonlinear least-squares to seismic inversion. This last data-fitting application does not satisfy our central convexity assumption but nonetheless illustrates the practical relevance of our approach to difficult problems.

Our numerical experiments compare the following three methods.

*Deterministic.* A conventional quasi-Newton linesearch method that uses the true function $f$ and gradient $\nabla f$ at every iteration. The method is based on a limited-memory BFGS Hessian approximation and a linesearch based on Hermite cubic-polynomial interpolation and the strong Wolfe conditions. Several comparison studies indicate that these type of limited-memory quasi-Newton methods are among the most efficient deterministic methods available for solving large-scale logistic-regression and CRF problems [19, 36, 21, 32].

*Stochastic.* An incremental-gradient method based on the iteration (1.2) with $g_k = \nabla f_i(x_k)$ and a constant $\alpha_k$, where the index $i \in \{1, \ldots, M\}$ is randomly selected. This corresponds to a constant sample size of one, and this simple method has proved competitive with more advanced deterministic methods like the one above for estimation in CRF models [35].

*Hybrid.* This is the proposed method described in section 4, which uses search directions computed from (1.6) and a linesearch based on satisfying condition (4.2). As with the deterministic method described above, the Hessian approximations are based on limited-memory BFGS, and the linesearch uses polynomial interpolation. The objective and gradient approximations are based on (4.1), where $\mathcal{B}_k \subseteq \{1, \ldots, M\}$,

and the number of elements in the current sample is initially 1 and grows linearly as per the formula

$$|\mathcal{B}_{k+1}| = \lceil \min\{1.1 \cdot |\mathcal{B}_k| + 1,\, M\} \rceil.$$

The hybrid nature of this approach should now be clear: the very first iteration is similar to the stochastic method; when the sample size grows to include all observations, the algorithm morphs into the deterministic method.

All experiments are carried out using MATLAB R2010b on a 64-bit Athlon machine. Two plots are shown for each experiment. The first shows the progress of the objective value against the index $p = \frac{1}{M} \sum_{k=0}^{p} |\mathcal{B}_k|$ for $p = 1, 2, \ldots$, which measures the effective number of passes through the entire data set. The second plot shows the cumulative number of $f_i$ functions evaluated, i.e., $\sum_{i=0}^{k} |\mathcal{B}_i|$, against the iterations $k$.

**5.1. Binary logistic regression.** Logistic regression models [5, section 4.3.2] are used in an enormous number of applications for the problem of *binary classification*. We are given data with $M$ examples of input-output pairs $(a_i, b_i)$, where $a_i \in \mathbb{R}^n$ is a vector of $n$ features, and $b_i \in \{-1, 1\}$ is a corresponding binary outcome. The goal is to build a linear classifier that, given the features $a_i$ and a vector of parameters $x$, the sign of the inner-product $a_i^T x$ gives $b_i$. The logistic model gives the probability that $b_i$ takes the value 1:

$$p_1(b_i = 1 \mid a_i, x) = \frac{\exp(a_i^T x)}{\exp(a_i^T x) + 1} = \frac{1}{1 + \exp(-a_i^T x)}.$$

(Typically a bias variable is added, but we equivalently assume that the first element of $x$ is set to one.) Thus, the probability that $b_i$ takes the value $-1$ is $[1 - p(b_i = 1 \mid a_i, x)]$. We can write these two cases compactly as

$$p_1(b_i \mid a_i, x) = \frac{1}{1 + \exp(-b_i a_i^T x)}.$$

This is the probability function $p$ used in (5.1).

The dominant cost in computing $f$ and its gradient is the cost of forming the matrix-vector products $Ax$ and $A^T y$ (for some $y$), where the $M$ rows of the matrix $A$ are formed from the vectors $a_i$. The Hessian is $\nabla^2 f(x) = A^T D A$, where $D$ is a diagonal with elements $p_1(b_i \mid a_i, x) \cdot [1 - p_1(b_i \mid a_i, x)]$, which lie in the range $(0, 0.25]$. The (nonnegative) eigenvalues of the Hessian are thus bounded above by $0.25\|A\|^2$ and are strictly positive if $A$ has full rank. Combined with 2-norm regularization, the resulting function $f$ satisfies the assumptions of section 1.2.

Our experiments for binary logistic regression are based on the TREC 2005 data set, which contains 823,470 binary variables describing the presence of word tokens in 92,189 email messages involved in the legal investigations of the Enron corporation [9]. The target variable indicates whether the email was spam or not. The data set was prepared by Carbonetto [8, section 2.6.5], and we set the regularization parameter $\lambda = 0.01$.

The results of this experiment are plotted in Figure 5.1, where we define the optimal value as the best value found across the methods after 150 effective passes through the data, and where for the stochastic method we plot the three step sizes (among powers of 10) that gave the best performance over the allotted iterations. Of course, in practice we will not know what step size optimizes the performance of the stochastic method, and the lack of sensitivity of the result to the initial step size is

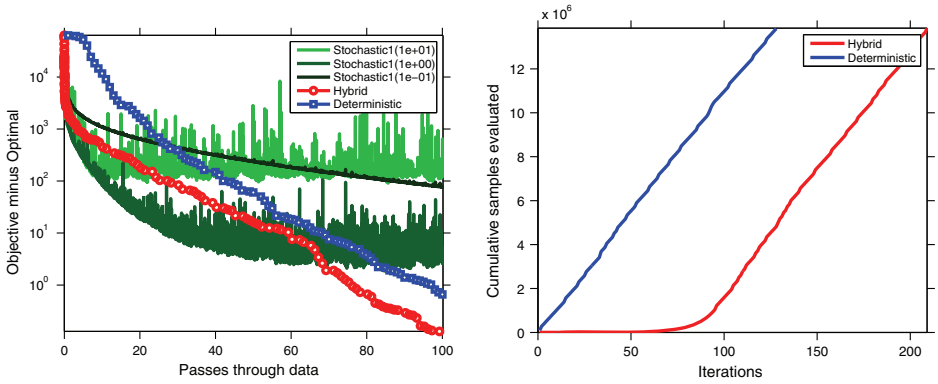



FIG. 5.1. *Binary logistic regression experiments for different optimization strategies for spam classification. The stochastic method is run with 3 different fixed steplengths.*

an advantage of the deterministic and hybrid methods. In the plot we see that, like the stochastic methods, the hybrid method makes rapid initial progress. Unlike the stochastic method, however, the hybrid method continues to make steady progress similar to the deterministic method. This behavior agrees with the theory.

**5.2. Multinomial logistic regression.** Multinomial logistic regression relaxes the binary requirement and allows each outcome $b_i$ to take any value from a set of classes $\mathcal{C}$ [5, section 4.3.4]. In this model there is a separate parameter vector $x_j$ for each class $j \in \mathcal{C}$. We model the probability that $b_i$ is assigned a particular class $j$ as

$$p_2(b_i = j \mid a_i, \{x_j\}_{j\in\mathcal{C}}) = \frac{\exp(x_j^T a_i)}{\sum_{j'\in\mathcal{C}} \exp(x_{j'}^T a_i)}. \tag{5.2}$$

This model is equivalent to binary logistic regression in the special case where the parameters $x_j$ of one class are fixed at zero and there are only two classes, i.e., $\mathcal{C} = \{-1, 1\}$. As with binary logistic regression, the function $p_2$ is log-concave and $-\log p_2$ has a Hessian whose eigenvalues are bounded above. Hence, the resulting function $f$ in (5.1) satisfies the assumptions of section 1.2.

Our experiments for multinomial logistic regression are based on the well-known MNIST data set [15], containing 70,000 examples of 28-by-28 images of digits, where each digit is classified as one of the numbers 0 through 9. The results are plotted in Figure 5.2, with $\lambda = 1$. The trends are similar to the binary logistic-regression experiments.

**5.3. Chain-structured CRFs.** The binary and multinomial logistic-regression models consider the case where a particular outcome $b_i$ is associated with a particular feature vector $a_i$. The CRF model takes multinomial logistic regression a step further by considering a set of feature vectors $a_i^k$ with which to predict corresponding values for discrete variables $b_i^k \in \mathcal{C}$, where $k$ takes values in a discrete ordered set $\Omega$.

We can naturally extend the multinomial logistic model to this scenario by defining the probability of a joint assignment $b_i$ as

$$p(\{b_i^k = j_k\}_{k\in\Omega} \mid \{a_i^k\}_{k\in\Omega}, \{x_j\}_{j\in\mathcal{C}}) = \frac{1}{Z_i} \prod_{k\in\Omega} \exp(x_{j_k}^T a_i^k).$$

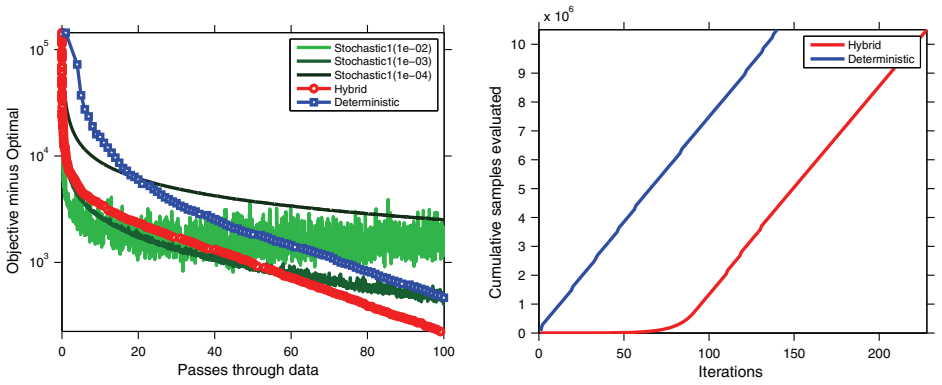



FIG. 5.2. *Multinomial logistic regression experiments for different optimization strategies for digit classification.*

(We assume that the parameter vectors $x_{j_k}$ are tied, so that $x_{j_k}$ is constant for all $k$; this assumption is not required in general.) The normalizing constant

$$Z_i = \sum_{j_1 \in \mathcal{C}} \sum_{j_2 \in \mathcal{C}} \cdots \sum_{j_k \in \mathcal{C}} \prod_{k \in \Omega} \exp(x_{j_k}^T a_i^k)$$

is chosen so that the distribution sums to one over all possible configurations of the $b_i^k$ variables. As written, computing $Z_i$ involves a very large number of terms, $|\mathcal{C}|^{|\Omega|}$; still, the sum can be computed efficiently by exchanging the order of operations. While this model is a straightforward generalization of multinomial logistic regression, it assumes that the labels in $\Omega$ that we are simultaneously predicting are independent. This might be unrealistic if, for example, the variables come from time-series data where $b_i^k$ and $b_i^{k+1}$ are likely to be correlated.

A chain-structured CRF [14] augments the model with additional terms that take into account sequential dependencies in the labels. It allows pairwise features $a_i^{kk'}$ and associated parameters $x_{kk'}$ and defines the probability of a configuration as

$$p_3(\{b_i^k = j\}_{k \in \Omega} \mid \{a_i^k\}_{k \in \Omega}, \{a_i^{kk'}\}_{k,k' \in \Omega, k'=k+1}, \{x_j\}_{j \in \mathcal{C}}, \{x_{j,j'}\}_{j,j' \in \mathcal{C}})$$
$$= \frac{1}{Z_i} \left[ \prod_{k \in \Omega} \exp(x_{j_k}^T a_i^k) \right] \cdot \left[ \prod_{\substack{k,k' \in \Omega \\ k'=k+1}} \exp(x_{j_k j_{k'}}^T a_i^{kk'}) \right].$$

The normalizing constant $Z_i$ is again set so that the distribution sums to one, and it can be computed using a variant on the forward-backward algorithm used in hidden Markov models [30, section III]. Because we need to run the forward-backward algorithm for each $i$, the probability function $p_3$ is significantly more expensive to evaluate than the corresponding multinomial probability function $p_2$; see (5.2).

For our chain-structured CRF experiments, we use the noun-phrase chunking problem from the CoNLL-2000 Shared Task [31], where the goal is to assign each word in a sentence to one of 22 possible states, and we use approximately 1.6 million features to represent the presence of words and word combinations [31, 32]. The CoNLL-2000 data set contains 211,727 words grouped into 8,936 sentences. The results of this experiment are plotted in Figure 5.3, where the regularization parameter is $\lambda = 1$.

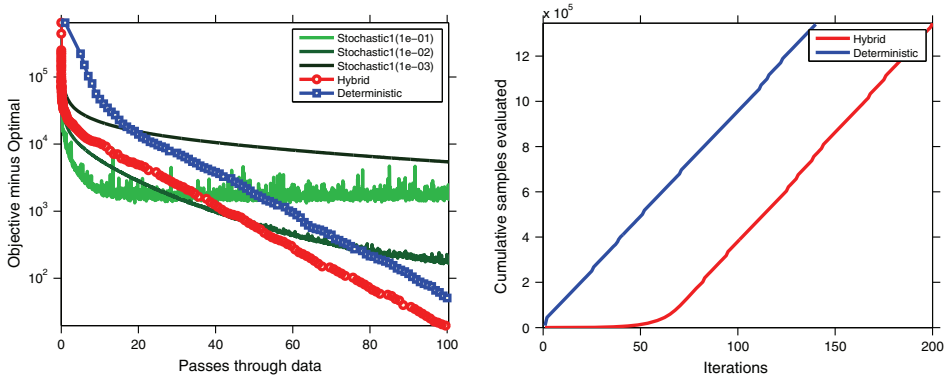



FIG. 5.3. *Chain-structured CRF experiments for different optimization strategies on the noun-phrase chunking task.*

**5.4. General CRFs.** While chain structures can model sequential dependencies in the labels, we might be interested in other structures, such as lattice structures for modeling image data. In order to use general CRFs [12, sections 4.6.1 and 20.3.2], we define a graph $\mathcal{G}$ where the nodes are the labels $\{1, 2, \ldots, w\}$ and the edges are the variable dependencies that we wish to consider. We then define the probability of $b_i$ taking a configuration $b_i^k = j_k$, for $k \in \Omega$, as

$$p_4(\{b_i^k = j_k\}_{k\in\Omega} \mid \{a_i^{k'}\}_{k'\in\Omega}, \{a_i^{kk'}\}_{(k,k')\in\mathcal{E}}, \{x_j\}_{j\in\mathcal{C}}, \{x_{jj'}\}_{(j,j')\in\mathcal{C}})$$
$$= \frac{1}{Z_i}\left[\prod_{k\in\Omega} \exp(x_{j_k}^T a_i^k)\right] \cdot \left[\prod_{(k,k')\in\mathcal{E}} \exp(x_{j_k j_{k'}}^T a_i^{kk'})\right].$$

In this general case, computing $Z_i$ is in the complexity class $\sharp P$, and the best-known algorithms have a runtime that is exponential in the tree-width of the graph $\mathcal{G}$ [12, sections 9–10]. For a two-dimensional lattice structure, the tree-width is the minimum between the two dimensions of the structure, so computing $Z_i$ is only feasible if one of the dimensions is very small (in the degenerate one-dimensional chain-structured case, the tree-width is one).

Because of the intractability of computing $Z_i$, we consider optimizing a pseudo-likelihood approximation [4] based on the probability model

$$p_4(\{b_i^k = j_k\}_{k\in\Omega} \mid \{a_i^k\}_{k\in\Omega}, \{a_i^{kk'}\}_{(k,k')\in\mathcal{E}}, \{x_j\}_{j\in\mathcal{C}}, \{x_{jj'}\}_{(j,j')\in\mathcal{C}})$$
$$\approx \prod_{k\in\Omega} p(b_i^k = j_k \mid \{a_i^{k'}\}_{k'\in\Omega}, \{a_i^{kk'}\}_{(k,k')\in\mathcal{E}}, \{x_j\}_{j\in\mathcal{C}}, \{x_{jj'}\}_{(j,j')\in\mathcal{C}}, \{b_i^{k'}\}_{k'\in\Omega,k'\neq k}).$$

The individual terms in this product of conditionals have the form of a multinomial logistic regression probability and are straightforward to compute. This is the function $p$ used to define the objective function $f$ in (5.1).

Our experiments on general CRFs are based on the image-denoising experiments described by Kumar and Hebert [13]. We use their set of 50 synthetic 64-by-64 images. Figure 5.4 shows the performance of the different methods with a regularization parameter of $\lambda = 1$. Figure 5.5 illustrates the marginal probabilities for the different methods at various points in the optimization for a randomly chosen image in the

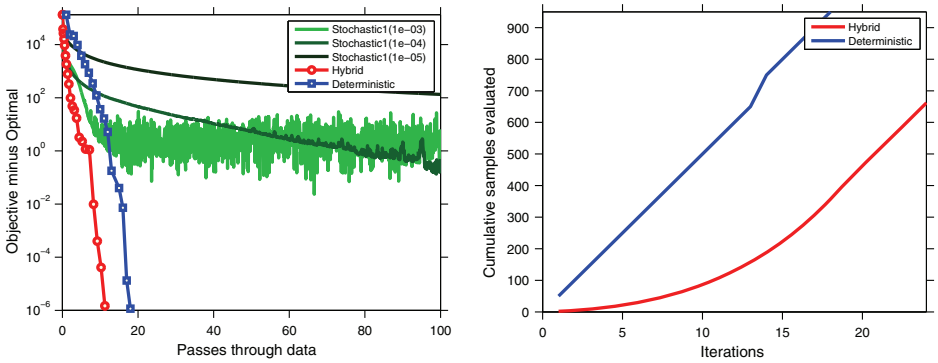



FIG. 5.4. *Lattice-structured CRF experiments for different optimization strategies for an image denoising task.*

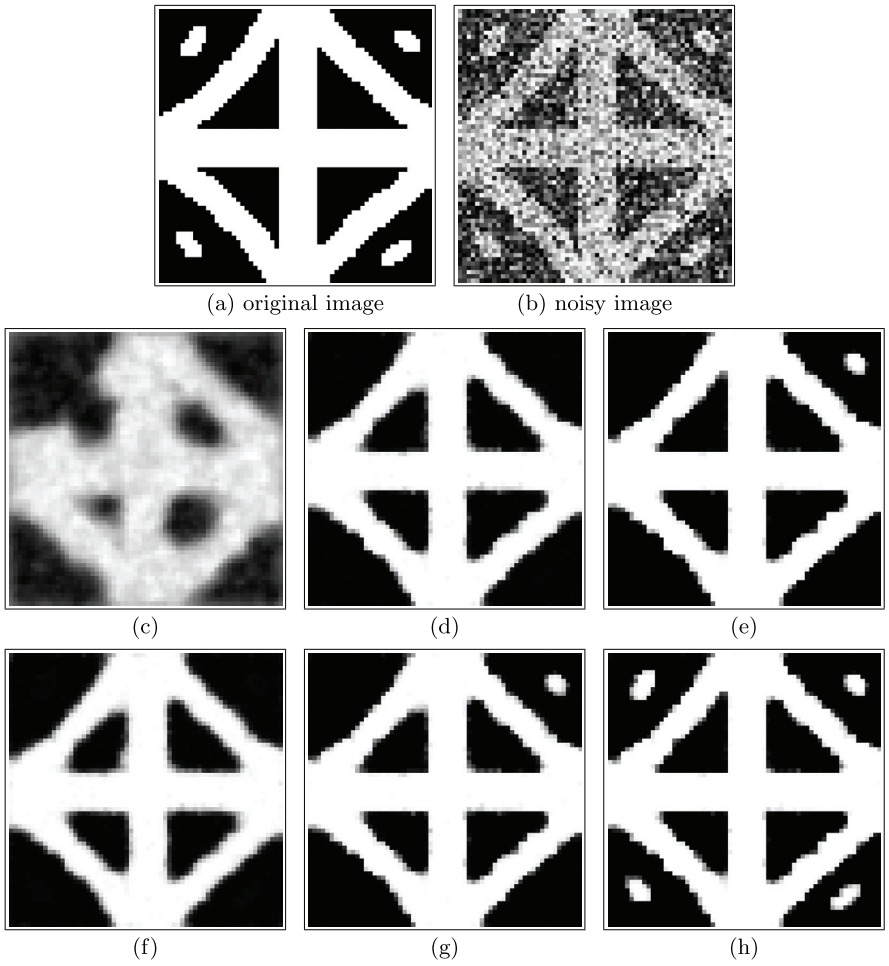



FIG. 5.5. *Top row: Original* (a) *and noisy* (b) *image. Second row: Marginals after two passes through the data for deterministic* (c), *stochastic* (d), *and hybrid* (e). *Third row: Marginals after five passes through the data for deterministic* (f), *stochastic* (g), *and hybrid* (h).

data set. (For the stochastic method, we plot the result with a step size of $\alpha = 10^{-4}$.) To approximate these marginals, we use the loopy belief propagation message-passing algorithm [5, section 8.4.7]. In these plots we see that the deterministic method does poorly even after two full passes through the data set, while the stochastic and hybrid methods do much better. After five passes through the data set, the hybrid method has found a solution that is visually nearly indistinguishable from the true solution, while it is still possible to see obvious differences in the deterministic and stochastic methods.

**5.5. Seismic inversion.** This last numerical experiment is a seismic inversion problem described by Van Leeuwen, Friedlander, and Herrmann [34]. The aim here is to recover an image of underground geological structures using only data collected by geophone receivers placed at the surface of the earth; these geophones record acoustic "shots" created by sources also at the surface.

The waveform inversion problem attempts to find a model $x$ of the subsurface structure that minimizes the nonlinear least-squares misfit as measured by the function

$$\phi(x) = \sum_{i=1}^{M} \sum_{\omega \in \Omega} \|d_i - P H_\omega(x)^{-1} q_i\|^2.$$

Each index $i$ corresponds to a particular shot (i.e., an observation) created by the source $q_i$ which creates a measurement $d_i$; each experiment samples a set of frequencies $\omega \in \Omega$. The matrix $P$ samples the wavefield at the receiver locations. The main cost in evaluating $\phi$ is solving the Helmholtz equation $H_\omega[x]u = q_i$ for each $i$, which is an expensive partial differential equation. Regularization is typically achieved by truncating the solution process [34]. Although the function $\phi$ is nonconvex and does not satisfy our standing assumptions, it hints at the applicability of the hybrid approach for solving difficult problems of important practical interest.

The results shown in Figure 5.6 are based on a relatively small two-dimensional example that involves $M = 101$ sources measuring eight frequencies. (Larger experiments, especially in three dimensions, are only feasibly accomplished on a computing cluster.) Because the problem is not strongly convex and hence we do not expect a fast convergence rate when far from this solution, for this problem the hybrid method increases the sample size by only one sample at each iteration, i.e.,

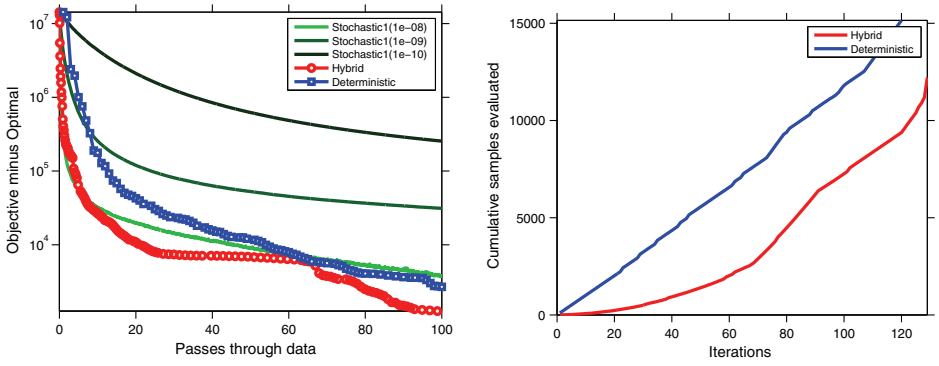



FIG. 5.6. *Nonlinear least-squares experiments for different optimization strategies on a seismic inversion problem.*

$|\mathcal{B}_{k+1}| = \min\{|\mathcal{B}_k| + 1, M\}$. After 20 passes through the data, the hybrid method clearly outperforms the deterministic method. The best stochastic method performs nearly as well as the deterministic and hybrid methods for the first 60 iterations, though with other step sizes it performs poorly. Although the figure shows the best methods all achieving similar residuals after 60 passes through the data, in practice that involves a prohibitive number of Helmholtz solves; practitioners are interested in making quick progress with as few solves as possible (and without needing to test a variety of step sizes to achieve good performance).

**6. Discussion.** Our work has focused on inexact-gradient methods for the unconstrained optimization of differentiable strongly-convex objectives. We anticipate that a similar convergence analysis with inexact gradients could be applied to other algorithms, such as Nesterov's accelerated gradient method [26, section 2.1]. In this case, it may be possible to relax the strong convexity assumption and obtain the optimal $\mathcal{O}(1/k^2)$ rate for an inexact-gradient version of this algorithm. The optimal $\mathcal{O}(1/k^2)$ rate using a gradient approximation whose error is uniformly bounded across iterations has been established by several authors, notably d'Aspremont [10]. But allowing a variable error would encourage more flexibility in the early iterations and would allow for eventually solving the problem to arbitrary accuracy. Although the $\mathcal{O}(1/k^2)$ rate is sublinear, it is substantially faster than the optimal $\mathcal{O}(1/\sqrt{k})$ achievable by methods that use noisy gradient information [24, section 14.1].

We might also obtain analogous rates for proximal-gradient methods for optimization with convex constraints or nondifferentiable composite optimization problems, such as 1-norm regularization [27]. The more general class of mirror-descent methods [1], which are useful for problems with a certain geometry such as optimization with simplex constraints, also seem amenable to analysis in our controlled error scenario.

We considered the case of bounded noise or noise that can be bounded in expectation and subsequently derived convergence rates and expected convergence rates, respectively. We might also consider the case where the noise is bounded with a certain probability. If the individual $\nabla f_i(x)$ are concentrated around $\nabla f(x)$, this might allow us to use concentration inequalities [20] to show that the convergence rates hold with high probability. Although we have analyzed an arbitrary strategy for selecting the elements of the sample and shown that uniform sampling achieves a better bound in expectation, it is possible that a quasi-random selection of the individual gradients might further refine the bound [22].

Although our emphasis here is on data-fitting applications where the error is a byproduct of subsampling the data, our analysis and implementation may be useful for other problems. For example, Gill, Murray, and Wright [11, p. 357] discuss the case of an objective function that can be evaluated to a prescribed accuracy (e.g., it could depend on an iterative process or a discretization level). They suggest solving the optimization problem over a sequence of tighter function accuracies. Our work provides a formal analysis and practical implementation of a method where the accuracy might be increased dynamically each iteration rather than solving a sequence of intermediate optimization problems. As a more recent example, Poyiadjis, Doucet, and Singh [29] consider approximating gradients in non-Gaussian state-space models using particle filters. Here, the variance of the approximation is directly proportional to the number of particles, and thus our work provides guidelines for selecting the number of particles to use in the approximation on each iteration.

**Acknowledgments.** Thanks to Kevin Swersky for the suggesting the growing sample-size strategy, and to Sasha Aravkin, Francis Bach, Hoyt Koepke, Pierre Jacob, Simon Lacoste-Julien, and Nicolas Le Roux for valuable discussions. We are grateful to two anonymous referees for their thoughtful comments, which led to many detailed changes, including the addition of section 2.3.

## REFERENCES

[1] A. Beck and M. Teboulle, *Mirror descent and nonlinear projected subgradient methods for convex optimization*, Oper. Res. Lett., 31 (2003), pp. 167–175.
[2] D. P. Bertsekas, *A new class of incremental gradient methods for least squares problems*, SIAM J. Optim., 7 (1997), pp. 913–926.
[3] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Nashua, NH, 1996.
[4] J. Besag, *Statistical analysis of non-lattice data*, The Statistician, 24 (1975), pp. 179–195.
[5] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
[6] D. Blatt, A. O. Hero, and H. Gauchman, *A convergent incremental gradient method with a constant step size*, SIAM J. Optim., 18 (2007), pp. 29–51.
[7] R. H. Byrd, G. M. Chin, W. Neveitt, and J. Nocedal, *On the use of stochastic Hessian information in optimization methods for machine learning*, SIAM J. Optim., 21 (2011), pp. 977–995.
[8] P. Carbonetto, *New Probabilistic Inference Algorithms that Harness the Strengths of Variational and Monte Carlo Methods*, Ph.D. thesis, University of British Columbia, Vancouver, Canada, 2009.
[9] G. V. Cormack and T. R. Lynam, *Spam corpus creation for TREC*, in Proceedings of the 2nd Conference on Email and Anti-Spam, available online from http://plg.uwaterloo.ca/∼gvcormac/treccorpus/, 2005.
[10] A. d'Aspremont, *Smooth optimization with approximate gradient*, SIAM J. Optim., 19 (2008), pp. 1171–1183.
[11] P. Gill, W. Murray, and M. Wright, *Practical Optimization*, Academic Press, New York, 1981.
[12] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, Cambridge, MA, 2009.
[13] S. Kumar and M. Hebert, *Discriminative fields for modeling spatial dependencies in natural images*, in Advances in Neural Information Processing Systems 16, MIT Press, Cambridge, MA, 2004, pp. 1531–1538.
[14] J. Lafferty, A. McCallum, and F. Pereira, *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*, in Proceedings of the 18th International Conference on Machine Learning, Morgon, Kauffmann, San Francisco, CA, 2001.
[15] Y. LeCun and C. Cortes, *The MNIST Database*, http://yann.lecun.com/exdb/mnist/, 1998.
[16] S. Lohr, *Sampling: Design and Analysis*, Duxbury Press, Pacific Grove, CA, 1999.
[17] D. Luenberger and Y. Ye, *Linear and nonlinear programming*, Springer-Verlag, New York, 2008.
[18] Z. Luo and P. Tseng, *Error bounds and convergence analysis of feasible descent methods: A general approach*, Ann. Oper. Res., 46 (1993), pp. 157–178.
[19] R. Malouf, *A comparison of algorithms for maximum entropy parameter estimation*, in Proceedings of the International Conference on Computational Linguistics, Taipei, Taiwan, 2002.
[20] P. Massart, *Concentration Inequalities and Model Selection*, Lecture Notes in Math. 1896, Springer, New York, 2007.
[21] T. Minka, *Algorithms for Maximum-Likelihood Logistic Regression*, Tech. rep., CMU, 2003.
[22] W. J. Morokoff and R. E. Caflisch, *Quasi-random sequences and their discrepancies*, SIAM J. Sci. Comput., 15 (1994), pp. 1251–1279.
[23] A. Nedic and D. Bertsekas, *Convergence rate of incremental subgradient algorithms*, in Stochastic Optimization: Algorithms and Applications, Kluwer, Dordrecht, The Netherlands, 2000, pp. 263–304.
[24] A. Nemirovski, *Efficient methods in convex programming*, Lecture notes, 1994.
[25] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM J. Optim., 19 (2009), pp. 1574–1609.

[26] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer, Dordrecht, The Netherlands, 2004.
[27] Y. Nesterov, *Gradient methods for minimizing composite objective function*, CORE Discussion Paper 76, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007.
[28] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.
[29] G. Poyiadjis, A. Doucet, and S. Singh, *Particle approximations of the score and observed information matrix in state space models with application to parameter estimation*, Biometrika, 98 (2011), pp. 65–80.
[30] L. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, Proc. of the IEEE, 77 (1989), pp. 257–286.
[31] E. Sang and S. Buchholz, *Introduction to the CoNLL-*2000 *Shared Task: Chunking*, in Proceedings of the Conference on Natural Language Learning, Lisbon, Portugal, 2000, pp. 127–132.
[32] F. Sha and F. Pereira, *Shallow parsing with conditional random fields*, in Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, 2003.
[33] D. F. Shanno and K.-H. Phua, *Matrix conditioning and nonlinear optimization*, Math. Program., 14 (1978), pp. 149–160.
[34] T. van Leeuwen, S. M, M. P. Friedlander, and F. Herrmann, *A hybrid stochastic-deterministic optimization method for waveform inversion*, in 73rd EAGE Conference & Exhibition, 2011.
[35] S. V. N. Vishwanathan, N. Schraudolph, M. Schmidt, and K. Murphy, *Accelerated training of conditional random fields with stochastic gradient methods*, in Proceedings of the International Conference on Machine Learning, Pittsburg, PA, 2006, pp. 969–976.
[36] H. Wallach, *Efficient training of conditional random fields*, Master's thesis, University of Edinburgh, Scotland, UK, 2002.