INFORMS TutORials in Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Stochastic Gradient Descent: Recent Trends

David Newton, Farzad Yousefian, Raghu Pasupathy

To cite this entry: David Newton, Farzad Yousefian, Raghu Pasupathy. Stochastic Gradient Descent: Recent Trends. *In* INFORMS TutORials in Operations Research. Published online: 19 Oct 2018; 193-220.
https://doi.org/10.1287/educ.2018.0191

https://pubsonline.informs.org/series/educ/

TUTORIALS IN OPERATIONS RESEARCH
 | ISBN 978-0-9906153-2-3
https://doi.org/10.1287/educ.2018.0191

Stochastic Gradient Descent: Recent Trends

David Newton,[a] Farzad Yousefian,[b] Raghu Pasupathy[a]
[a] Department of Statistics, Purdue University, West Lafayette, Indiana 47907; [b] Industrial Engineering and Management, Oklahoma State University, Stillwater, Oklahoma 74078

Contact: newton34@purdue.edu (DN); farzad.yousefian@okstate.edu (FY); pasupath@purdue.edu (RP)

Abstract Stochastic gradient descent (SGD), also known as *stochastic approximation*, refers to certain simple iterative structures used for solving stochastic optimization and root-finding problems. The identifying feature of SGD is that, much like gradient descent for deterministic optimization, each successive iterate in the recursion is determined by adding an appropriately scaled gradient estimate to the prior iterate. Owing to several factors, SGD has become the leading method to solve optimization problems arising within large-scale machine learning and "big data" contexts such as classification and regression. In this tutorial, we cover the basics of SGD with an emphasis on modern developments. The tutorial starts with stochastic optimization examples and problem variations where SGD is applicable, and then it details important flavors of SGD that are currently in use. The oral presentation of this tutorial will include numerical examples.

Keywords stochastic gradient descent • stochastic approximation • stochastic optimization • machine learning • optimization • big data

1. Introduction and Preliminaries

This tutorial considers unconstrained smooth stochastic optimization problems having the form

$$\begin{aligned} \min \ & f(x) = \mathbb{E}[F(x)] = \int_{\Xi} F(x,\xi) P(d\xi) \\ \text{s.t.} \ & x \in X \subset \mathbb{R}^n. \end{aligned} \tag{P}$$

The objective function f is assumed to be bounded from below—that is, $\inf_{x\in X} f(x) > -\infty$; this tutorial covers methods for both smooth and nonsmooth functions f. Unlike deterministic optimization (Nocedal and Wright [74]), the function f is not directly observable, but we have access to a "first-order stochastic oracle" that can be "called" to obtain unbiased estimates $F(x,\xi)$ and $G(x,\xi)$ of $f(x)$ and $\nabla f(x)$, respectively, at any requested point $x \in X$. We assume that $F(x,\cdot)$ and $G(x,\cdot)$ are unbiased estimates of $f(x)$ and $\nabla f(x)$; that is, $f(x) = \mathbb{E}_{\xi}[F(x,\xi)]$, $\nabla f(x) = \mathbb{E}_{\xi}[F(x,\xi)]$.

The problem statement that appears in (P) has recently generated renewed interest because of its direct applicability in parameter estimation problems arising within modern machine learning settings such as regression, classification, clustering, and anomaly detection (Hastie and Tibshirani [44]). As some observers have noted (Bottou et al. [15]), deep neural networks as a modeling paradigm, in concert with efficient stochastic optimization algorithms (mainly, stochastic gradient descent to solve problem P), have resulted in spectacular successes in diverse domains.

Especially because the importance of problem P has been recorded at length elsewhere, we will not devote any more space to motivating problem P. Instead, this tutorial will cover

modern solution methods, particularly variants of *stochastic approximation* (Borkar [14], Kushner and Yin [53]), for solving problem P. Stochastic approximation, more recently called *stochastic gradient descent* (SGD), is understood here as recursions having the form

$$x_{k+1} = \Pi_X\big(x_k - \alpha_k H_k^{-1} g(x_k, M(x_k))\big), \tag{SGD}$$

where the term α_k appearing in (SGD) is called the *step size*, the positive definite symmetric matrix H_k approximates the Hessian matrix of second derivatives $H(\cdot)$ at x_k, and $g(x_k, M(x_k))$ approximates the gradient $\nabla f(x_k)$ using sample size $M(x_k)$. Traditionally, the sample size $M(x_k)$ is set to some fixed value, but recent work has questioned such choice.

1.1. Some Terminology and Notions

The following terminology and notions will be assumed throughout the tutorial.

1. *Solving the problem P*: An iterative algorithm will have "solved" the problem P if it generates a stochastic sequence $\{x_k\}$ that satisfies $\|\nabla f(x_k)\| \to 0$ as $k \to \infty$ with probability 1 (w.p.1.) or if $\mathbb{E}[\|\nabla f(x_k)\|] \to 0$ as $k \to \infty$. Of course, this places no guarantees on the behavior of the sequence $\{x_k\}$ itself without further structural assumptions on the function f. Furthermore, the function f may have multiple minima (or none), and a guarantee such as $\|\nabla f(x_k)\| \to 0$ w.p.1 says little about which, if any, of the local minima of f are attained.

2. *Nature of the "stochastic oracle"*: The notion of a "stochastic oracle" is deliberately left vague to subsume a variety of contexts. For the purposes of this tutorial, a *stochastic oracle* is either a Monte Carlo simulation (Asmussen and Glynn [1], Glasserman [39], Nelson [67]) or a large data set containing (virtually) an infinite number of observations. Accordingly, "calling the stochastic oracle" at the point $x \in \mathbb{R}^p$ using the "random seed" ξ_i results in observing the function estimate $F(x, \xi_i)$ and the gradient (or subgradient) estimate $G(x, \xi_i)$.

3. *Algorithm assessment and work complexity*: The number of calls to the stochastic oracle is the sole unit of computational burden. Thus, the work complexity for the purposes of this tutorial relates to the quality of a solution returned by an algorithm as a function of the number of calls to the stochastic oracle. As an example, when we say that an algorithm exhibits $\mathcal{O}(\epsilon^{-1})$ complexity (on a nonconvex smooth objective function), we mean that the solution x_k returned after k calls to the stochastic oracle guarantees that $\mathbb{E}[\|\nabla f(x_k)\|^2] \le \epsilon$. On the other hand, the *iteration complexity* of an iterative algorithm refers to the quality of a solution returned by an algorithm as a function of the number of iterations undertaken by the algorithm. Iteration complexity is not useful as a measure in the current context except when the sample size during each iteration is fixed, in which case the work complexity and the iteration complexity differ only by a constant.

1.2. Scope of the Tutorial

This tutorial is primarily aimed at early researchers and consumers of stochastic optimization. The content will accordingly be kept at an accessible level. Codes corresponding to the best-performing algorithms included in the tutorial will be available through a website that will be advertised during the oral presentation of this tutorial or from the author upon request.

Because stochastic optimization has recently become all encompassing, we find it especially necessary to list caveats and key topics that this tutorial will *not cover*. This is a tutorial on the recent variants of SGD that, for the purposes of this tutorial, is understood to have the structure in (SGD). Numerous other paradigms that have recently gained prominence, such as stochastic trust region methods (Bandeira et al. [2], Chen et al. [23], Shashaani et al. [89, 90], sample-average approximation (Shapiro et al. [88]), and retrospective approximation (Pasupathy [76], Pasupathy and Schmeiser [77]), will not be discussed here.

As has been noted, SGD is the new nomenclature for stochastic approximation that was first introduced through a seminal paper by Robbins and Monro [82], followed by a flurry of important papers (Chung [24], Fabian [31], Kiefer and Wolfowitz [51], Sacks [84], Venter [97])

on various aspects. Such interest has continued over the previous six decades, and an enormous literature that includes comprehensive surveys (Kushner and Yin [53], Lai [54]) on virtually all aspects of stochastic approximation has been written. Wisely, this tutorial will not undertake to supplant or add to any of these surveys on stochastic approximation. Instead, the tutorial explains trends in the last decade to two decades appearing under the topic *stochastic gradient descent* or *stochastic approximation*. Such papers have appeared across different research subcultures but seem to have particular relevance to machine learning and big data contexts. And whereas most of these more recent methods have already appeared in some form in the older stochastic approximation literature, they tend to have an increased focus on complexity results afforded through more stringent structural assumptions on the function f.

Owing to space restrictions here, and time restrictions on the oral presentation, solutions to several important variations of problem P will not be considered in the tutorial.

1. This tutorial will discuss only algorithms that assume access to a first-order oracle—that is, the algorithms that have access to "direct gradient (or subgradient)" observations $G(x, \xi)$. A substantial number of simulation contexts are "derivative-free," implying that even though the gradient $\nabla f(x)$ might exist at the point $x \in \mathbb{R}^p$, estimates of $\nabla f(x)$ can be constructed only using methods such as finite differencing (Asmussen and Glynn [1]). Similarly, optimization in problem P is stated to be on finite-dimensional Euclidean spaces, excluding important stochastic optimization problems on non-Euclidean spaces (Bubeck [17], Nemirovskii et al. [68]).

2. A substantial fraction of methods that appear under the SGD banner are conditioned on a given data set. One of the many important implications of such conditioning is that any mathematical expectation that appears in a claimed result on complexity is on the probability space on which *the algorithm* is defined and *conditional on the given data set.* Optimization in such contexts has sometimes been called *empirical risk minimization* (Bottou et al. [15]). Three prominent examples of algorithms in this category are Le Roux et al. [57], Johnson and Zhang [50], and Defazio et al. [28]. The fact that algorithmic inference in such contexts is conditional on the given data set is often missed or ignored, leading to misinterpretations about the effectiveness of an algorithm. Because our interest is on inference made at the population level, we have tended to deemphasize algorithms that are customized to a specific data set.

3. The SGD literature contains a substantial fraction of algorithms that apply to the *online learning* context. Such problems are closely related to stochastic optimization problems (Bubeck [16], Shalev-Shwartz and Tewari [87] but differ mainly in the way performance is measured. In the online learning setting, for example, algorithms seek to minimize *regret* measured as the deviation of the incumbent solution's quality from the optimal value, integrated over a specified time horizon. Again, because of space restrictions, online learning algorithms are not discussed in this tutorial.

4. An important class of methods, collectively called *momentum methods*, has been successful in solving (deterministic) convex optimization problems. The first versions of these algorithms, introduced in Polyak [102] and Nesterov [72], have now been extended to the stochastic convex and nonconvex contexts (Ghadimi and Lan [38], Yang et al. [103]. Momentum methods are extremely competitive and relevant to solving problem P discussed in this paper. After much deliberation, we have chosen not to discuss these methods because the current tutorial is on SGD, and momentum methods depart from the structure of the SGD iteration, even if it may be argued that such departure is minimal.

5. A popular alternative to SGD for solving problem P is sample-average approximation (SAA) (Kim et al. [51], Shapiro et al. [89]). Given the focus of the tutorial, SAA methods are not covered.

2. Two Motivating Contexts

In what follows, we provide two example contexts where the solution methods we discuss in this tutorial are relevant. The first example context is typical of machine learning settings involving a large amount of data that facilitate the construction of the relevant estimators. In the second example, by contrast, the "data generation" results from using Monte Carlo.

2.1. Example 1: Classification and Regression

Consider the context of classifying (or labeling) an object based on its observed features. An especially apt example of such object classification is that of face recognition using a photograph that is represented as a large number of pixels, each of which has a color value. This task is to be accomplished algorithmically by constructing a parametrized model that is trained over a given data set of photographs to minimize a chosen loss function.

To abstract this setup, suppose $W \in \mathcal{W} \subset \mathbb{R}^{d_w}$ and $Y \in \mathcal{Y}$ are well-defined random objects in a probability space that represent the *feature* and the *label* in machine learning parlance, respectively. Also, let $x \in \mathbb{R}^d$ denote the decision variable representing a parameter vector of interest, let $m(\cdot; x) : \mathcal{W} \to \mathcal{Y}$ denote a family of models parameterized by x, and let $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ denote a chosen "loss" function.

In the facial recognition scenario, the set $\mathcal{Y}$ could be a finite set $\{1, 2, \ldots, p\}$ of integers representing a fixed group of people to be identified, and the set $\mathcal{W} = \{1, 2, \ldots, N\} \times \{1, 2, \ldots, N\} \times \{0, 1, 2, \ldots, 255\}$, where a photograph has $N \times N$ pixels each taking a color value in the set $\{0, 1, 2, \ldots, 255\}$. A popular choice for the model $m(\cdot; \cdot)$ is the *linear model*:

$$m(w; x) = x^T \tilde{w}, \quad \tilde{w}^T = \left(w^T, 1\right).$$

And popular choices for the loss function $\ell(\cdot, \cdot)$ include the logistic loss $\ell(y', y) = \log(1 + \exp(-yy'))$, the hinge loss $\ell(y', y) = \max(0, 1 - yy')$, and the zero-one loss $\ell(y', y) = \mathbb{1}_{\{y' \neq y\}}$.

The stochastic optimization problem is then to identify the parameters $x \in \mathbb{R}^d$ that minimize the expected loss, where the expectation is taken with respect to the random objects (W, Y). Formally, we would like to solve

$$\begin{aligned} &\min\ f(x) = \mathbb{E}[\ell(m(W; x), Y)] = \int_{\mathcal{W} \times \mathcal{Y}} \ell(m(w; x), y) \mathbb{P}(d(w, y)) \\ &\text{s.t. } x \in \mathbb{R}^d. \end{aligned} \tag{2}$$

Of course, the objective function f cannot be directly observed. Instead, it can be estimated by using observed data (W_i, Y_i), $i = 1, 2, \ldots, n$ and an implicit or explicit model for the probability measure $\mathbb{P}$.

The difficulty of solving the optimization problem in (2) depends on the choices made for the model m and the loss ℓ. Specifically, it depends on whether these choices result in a convex optimization problem and whether direct gradients or subgradients are easily available. For example, the use of a hinge loss ℓ results in a convex objective function f, but the use of a zero-one loss results in a nonconvex objective. Similarly, deep neural networks, as noted in Bottou et al. [15], are essentially highly nonlinear and nonconvex models m that lend themselves to the easy construction of direct gradients through *back propagation*.

The stochastic optimization problem formulated in (2) using a parametrized model and a loss function subsumes a large number of other contexts including regression, compressed sensing, and matrix completion in numerous real-world scenarios. See Bubeck [16] for more on this issue.

2.2. Example 2: Portfolio Optimization

The classical Markowitz portfolio optimization problem (Markowitz [65]) in finance seeks proportions $x = (x_1, x_2, \ldots, x_d)$ of a given budget to be allocated across d assets in a given financial portfolio to maximize a combination of the expected return and variance over a given time horizon

$[0, T]$. The asset price movement over the time interval $[0, t]$ is assumed to be governed by a probability model such as, for example, geometric Brownian motion (Glasserman [39]).

Formally, suppose $Z_t \in \mathbb{R}^d, t \in [0, T]$ represents the asset price process for d assets in a portfolio, and suppose $\eta > 0$ is a *risk aversion* parameter for a user. Then, the Markowitz portfolio stochastic optimization problem can be written as

$$\min f(x) = \mu^T x - \eta\, x^T \Sigma\, x,$$
$$\text{s.t. } \sum_{i=1}^{d} x_i = 1, x_i \geq 0,$$

where $\mu = \mathbb{E}[Z_T]$ and $\Sigma = \mathrm{Var}(Z_T)$ are the mean and covariance of Z_T, respectively.

Because the returns Z_t are usually the result of detailed models of evolution of an asset over time (e.g., see chapter 3 in Glasserman [39]), the quantities μ and Σ are not known in closed form but estimators $\hat{\mu}_n$, $\hat{\Sigma}_n$ of μ and Σ, respectively, can be constructed using Monte Carlo. In fact, because the function f is quadratic and concave, estimators for its first derivative $\mu - 2\eta\Sigma x$ and second derivative $2\eta\Sigma$ are readily available.

The portfolio optimization problem exemplifies numerous contexts where the underlying objective function is very well behaved and lends itself to the easy construction of unbiased derivative estimators that are of immense value within algorithms for stochastic optimization. This is in contrast to many simulation settings where such derivative estimators are not available directly.

3. Basic SGD

The basic SGD algorithm (Robbins and Monro [82]), mimicking gradient descent (Nocedal and Wright [74]) for deterministic optimization, proceeds by taking a step of size α_k along the negative gradient estimate $-g(x_k, 1)$. As the notation makes explicit, the gradient estimate $g(x_k, 1)$ is obtained using a unit sample size. A formal listing of basic SGD appears in Algorithm 1.

Algorithm 1 (Basic SGD)

1. Initialize x_0
2. Obtain stochastic gradient $g(x_k, 1)$
3. Set $x_{k+1} = \Pi_X(x_k - \alpha_k g(x_k, 1))$, where $\Pi_X(z) = \arg\inf_{x \in X}\{\|z - x\|\}$.
4. Set $k = k + 1$
5. Go to step 1

The following result characterizes the convergence of the iterates generated by Algorithm 1 under the assumption that the function $F(\cdot, \xi)$ is convex.

Theorem 1. *In problem P, let the set X be convex and closed, let the optimal set X^* of problem P be nonempty, and let the function $F(\cdot, \xi)$ be convex on $\mathbb{R}^n$ for each ξ. Also, let the function f be L-smooth; that is, f is differentiable with gradient $\nabla f(\cdot)$ satisfying $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^n$. Furthermore, let the following additional assumptions hold:*

(1) *The step sizes $\alpha_k \geq 0$ satisfy $\sum_{k=1}^{\infty} \alpha_k = \infty$, $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$;*
(2) $\sum_k \alpha_k^2\, \mathbb{E}[\|g(x_k, 1) - \nabla f(x_k)\|^2 | \mathcal{F}_k] < \infty$.

Then, $\lim_k \inf_{x \in X^*}\{\|x_k - x\|\} = 0$ *w.p.*1.

Theorem 1 appears in Yousean et al. [104]; a proof of Theorem 1 follows using simple arguments that are standard in the analysis of optimization algorithms for convex smooth functions.

Three aspects of the result in Theorem 1 are noteworthy. First, for the iterates generated by Algorithm 1 to converge, the step size α_k has to diminish to zero, and slow enough, as encoded in the assumption appearing in item (1) of Theorem 1. This is a clear departure from the deterministic context where under similar conditions on f, the gradient algorithm with fixed

step will converge to the optimal set X^* if the step size is small enough. Second, as noted in Yousean et al. [104], the assumption appearing in item (2) of Theorem 1 is satised if, for example, the error $\|g(x_k,1)-\nabla f(x_k)\|$ is uniformly bounded by a constant. However, a uniform bound on $\|g(x_k,1)-\nabla f(x_k)\|$ is often violated because the error $\|g(x_k,1)-\nabla f(x_k)\|$ is frequently proportional to $\|\nabla f(x_k)\|$. To cover such cases, a generalization of Theorem 1 using an assumption on the "spatial growth" of the error $\|g(x_k,1)-\nabla f(x_k)\|$ is obtainable using the method outlined in Polyak [78]. Third, we emphasize that Theorem 1 assumes that the sample-path functions $F(\cdot,\xi)$ are convex for each ξ. Numerous variations that do not assume the convexity of $F(\cdot,\xi)$ are available through standard references on stochastic approximation (e.g., Kushner and Yin [53]).

We next provide a result that provides a path to proving the convergence rate of SGD methods. We list this result assuming that the function f is λ-strongly convex.

Theorem 2. *In problem P, let the set X be convex and closed, and let the function $F(\cdot,\xi)$ be convex on $\mathbb{R}^n$ for each ξ. Also, let the function f* be *L-smooth and λ-strongly convex with a unique minimum attained at $x_* \in X$. Furthermore, let the following additional assumptions hold:*

(1) *The step sizes $\alpha_k \geq 0$ satisfy $\sum_{k=1}^{\infty}\alpha_k=\infty$, $\sum_{k=1}^{\infty}\alpha_k^2<\infty$;*
(2) *$\mathbb{E}[\|g(x_k,1)-\nabla f(x_k)\|^2|\mathcal{F}_k]<\nu^2$ for all k.*

Then, $\lim_k \mathbb{E}[\|x_k-x_*\|^2]=0$, and for every $\epsilon>0$,

$$\mathbb{P}\left(\|x_j-x_*\|\leq\epsilon \text{ for all } j\geq k\right)\leq 1-\frac{1}{\epsilon}\left(\mathbb{E}\left[\|x_k-x_*\|^2\right]+\nu^2\sum_{i=k}^{\infty}\alpha_i^2\right). \tag{3}$$

The condition $\sum_{k=1}^{\infty}\alpha_k=\infty$, $\sum_{k=1}^{\infty}\alpha_k^2<\infty$ on the step sizes that appear in Theorems 1 and 2 is satisfied by the choice $\alpha_k=c/k$, $c>0$. In fact, it can be shown using (3) that the choice $\alpha_k=c/k$, $c>0$ attains the fastest possible convergence rate to within a constant factor, as long as $2c\lambda>1$. Basic SGD is important for having laid down a simple algorithmic framework. However, it was quickly realized that, depending on the extent of "noise," even the optimal step size choice $\alpha_k=c/k$ might result in steps that are "too short." Perhaps more important, it was realized that SGD simply did not have a recipe for choosing an appropriate value of c. Implementers thus executed SGD with a fixed step size $\alpha_k=\alpha$, and not so surprisingly, such an algorithm does not converge to a stationary point. Instead, the typical behavior involved a rapid descent to some vicinity of a first-order stationary point of f, followed a random walk around the first-order critical point. The following theorem, described and proved in Bottou et al. [15], characterizes the expected behavior of SGD with fixed step size.

Theorem 3. *Suppose the objective function $f:\mathbb{R}^d\to\mathbb{R}$ is L-smooth and λ-strongly convex with unique minimum $x^*\in\mathbb{R}^p$. Also, let $\mathbb{E}[\|g(x_k,1)\|^2]\leq M+M_g\|\nabla f(x_k)\|^2$ and $\mathbb{E}[g(x_k,1)^T\nabla f(x_k)]\geq\mu\|\nabla f(x_k)\|^2$ for some $M,M_g,\mu>0$. If SGD is executed with the xed step size $\alpha_k=\alpha$ satisfying $0<\alpha\leq\lambda/LM_g$, then SGD has the iteration complexity*

$$\mathbb{E}[f(x_k)-f(x^*)]\leq\frac{\alpha LM}{2\lambda\mu}+(1-\alpha\lambda\mu)^{k-1}\left(f(x_1)-f(x^*)-\frac{\alpha LM}{2\lambda\mu}\right).$$

Notice that Theorem 3 implies that the iterates approach x^* exponentially fast (in k) but do not make any further progress. This makes intuitive sense, as a large step size makes it impossible to approach the point x^* beyond a fixed distance in any systematic manner. The analogue of Theorem 3 for diminishing step sizes α_k is stated as follows. A "process convergence" version of the above theorems characterizes convergence to an Ornstein–Uhlenbeck process (Asmussen and Glynn [1]) under certain scaling.

Theorem 4. *Suppose the objective function $f:\mathbb{R}^d\to\mathbb{R}$ is L-smooth and λ-strongly convex with unique minimum $x^*\in\mathbb{R}^p$. Also, let $\mathbb{E}[\|g(x_k,1)\|^2]\le M+M_g\|\nabla f(x_k)\|^2$ and $\mathbb{E}[g(x_k,1)^T\nabla f(x_k)]\ge\mu\|\nabla f(x_k)\|^2$ for some $M,M_g,\mu>0$. Then, SGD executed with the step size $\alpha_k=\beta/(\gamma+k)$, where $\beta\lambda\mu>1$, $\alpha_1 L M_g\le\mu$, and $\gamma>0$ has the iteration complexity*

$$\mathbb{E}[f(x_k)-f(x^*)]\le\frac{\max\left\{\frac{\beta^2LM}{2(\beta\lambda\mu-1)},(\gamma+1)(f(x_0)-f(x^*))\right\}}{\gamma+k}.$$

As noted earlier, the complexity $\mathcal{O}(1/k)$ reported by Theorem 4 is optimal "to within a constant" (see Section 6). As was commented earlier, unfortunately, such optimality does not guarantee good practical performance, especially because its choice in basic SGD takes no cognizance of the initial optimality gap $f(x_0)-f(x^*)$. In other words, a poor starting point x_0 demands a large β, whereas a good starting point will probably benefit more from a small β. Also notice from Theorem 4 that the ratio L/c, called the condition number of the function f, decides the complexity in a crucial way, echoing the deterministic gradient method (Nesterov [71]). Large L/c values imply poor conditioning and therefore more opportunity for SGD to take steps that are not productive in the sense of decreasing the objective function value.

3.1. Mini-Batch SGD

Recall that the context of problem P is such that we have access only to "noisy" gradients. This implies then, roughly speaking, that more noise in the observations should lead to more unproductive steps by SGD, leading one to wonder whether there are benefits to using larger samples at each iteration. In other words, instead of just using $g(x_k,1)$, we could employ a *mini-batch* SGD, where at each iteration we use a sample average of m noisy gradient estimates. Specifically, mini-batch SGD is the basic SGD iteration that uses the gradient approximation $g(x_k;m)$. A basic version of the mini-batch algorithm appears as Algorithm 2.

Algorithm 2 (Mini-Batch SGD)

1: Initialize x_0
2: Choose batch size $m\in\mathbb{N}$
3: Generate $\xi_i, i=1,2,\ldots,m$ independently and construct $g(x_k,m)$
4: Set $x_{k+1}=\Pi_X(x_k-\alpha_k g(x_k,m))$
5: Go to step 3

Given Theorem 4, the complexity of the mini-batch algorithm, easy to guess, is

$$\epsilon_k\le\frac{\max\{\frac{\beta^2LM}{2m(\beta\lambda\mu-1)},(\gamma+1)(f(x_0)-f(x^*))\}}{\gamma+k}.$$

It should be noted that although the mini-batch SGD is used widely, it is of little value over and above basic SGD when the mini-batch size m is not chosen based on any available information on the constant M. Practical implementations will usually involve a pilot run intended to obtain a rough sense of M, followed by an appropriate choice of m. In fact, most practical implementations of mini-batch SGD involve some sort of manual updating of the mini-batch size m. As noted in Bottou et al. [15], the real advantages of mini-batch SGD exist in a distributed or parallel computing context, where the massive parallelism of the mini-batch average computation can be exploited.

4. Balancing Bias and Variance with Aggregation

A key issue that governs the efficiency of SGD is the extent of sampling to obtain gradient estimates. The rudimentary SGD, outlined in Section 3, leans on one extreme where during each iteration, a gradient estimate is obtained using exactly one observation. That is, the

sample size during each iteration of SGD is set to unity. The mini-batch SGD, outlined in Section 3.1, in an attempt to correct inefficiencies in SGD as a result of the choice of sample size, sets the sample size during each iteration to $m > 0$, where m is some fixed positive integer that needs to be chosen.

A natural question to ask in response to the strategy adopted in mini-batch SGD pertains to *dynamic* and *adaptive* sampling. Specifically, why should sample size be fixed across iterations? Why not choose sample sizes in response to the proximity to a critical point? Although not immediately obvious, the underlying issue in the context of sample size choice is really that of balancing bias and variance of generated iterates. To see this more clearly, consider the squared L_2 norm $\mathbb{E}\|\nabla f(X_k)\|_2^2$ decomposed (orthogonally) into its squared bias and variance components:

$$\mathbb{E}\|\nabla f(X_k)\|_2^2 = \underbrace{(\mathbb{E}[\nabla f(X_k)])^T(\mathbb{E}[\nabla f(X_k)])}_{\text{squared bias}} + \underbrace{\mathrm{Tr}(\mathrm{Var}(\nabla f(X_k)))}_{\text{trace of variance}}. \tag{4}$$

If $\mathbb{E}\|\nabla f(X_k)\|_2^2$ converges to zero as $k \to \infty$, then each of the two component sequences in (4) has to converge to zero. And, interestingly, it can be shown that the rate at which the squared bias decays to zero depends on the chosen step size and not the sample size used to construct the gradient estimate, with steps that are "too small" resulting in slow convergence. The variance, on the other hand, depends on both the chosen step sizes and the chosen sample sizes, with larger step sizes or small sample sizes resulting in noisy SGD trajectories. So the efficiency of SGD is crucially dependent on the management of the two sequences in (4) by controlling step sizes and the sample sizes. In particular, the step sizes and the sample sizes should be chosen so that the two sequences appearing in (4) converge to zero "in lock-step." Dynamic and adaptive sampling methods, iterate-averaging methods, and gradient averaging methods can all be seen as attempting to balance the squared bias and variance terms in (4) in different ways.

Remark 1. The phrases *adaptive sampling* (*Bollapragada et al.* [13], *Hashemi et al.* [42, 43]), *dynamic sampling* (*Byrd et al.* [18]), *variable sample size* (*Deng and Ferris* [29], *Homem-de-Mello* [47]), *retrospective approximation* (*Chen and Schmeiser* [22], *Pasupathy* [76], *Pasupathy and Schmeiser* [77]), and *stochastic decomposition* (*Higle and Sen* [46, 47]) have all been used in the literature to reect similar if not identical but context-specic ideas. In this paper, we have been careful to use only the two phrases *dynamic sampling* and *adaptive sampling*. The former phrase is used to describe a sampling strategy in SGD iterations that uses a xed sequence of sample sizes that is a function of the iteration number. Adaptive sample sizes, by contrast, do not use a predetermined sequence; the sample sizes are instead determined on the y and are a function of the observed algorithm trajectory. Adaptive sample sizes are thus random, whereas dynamic samples sizes are xed. The theory pertaining to adaptive sampling is in its infancy as of this writing.

4.1. Dynamic and Adaptive Sampling Methods

Dynamic sampling methods use a prespecified sequence of sample sizes $\{m_k\}$ across SGD iterations. The sample sizes generally diverge—that is, $m_k \to \infty$ as $k \to \infty$—to reflect the need for better estimation as the iterates tend closer to a critical point. Discounting retrospective approximation (Kim et al. [53], Pasupathy [76]) and variable sample sizing ideas (Bayraksan and Morton [3], Deng and Ferris [29], Homem-de-Mello [47]) detailed in other contexts, the earliest mention of dynamic sampling appears to be in Bertsekas and Tsitsiklis [10]. This idea is developed to a fuller extent in Byrd et al. [18] and Friedlander and Schmidt [36]. In Byrd et al. [18], for instance, the dynamic sampling gradient algorithm (see Algorithm 3) is presented where the sample size m_k during the kth iteration of the SGD iteration is obtained as the smallest sample size such that the estimated standard error of the estimated gradient is no

more than the product of a constant θ and the norm of the estimated gradient. This sample sizing rule is formally stated as

$$m_k = \arg\inf\left\{|\mathcal{S}| : \frac{\sqrt{\left\|\hat{\mathrm{Var}}(g(x_k, |\mathcal{S}|))\right\|_1}}{\sqrt{|\mathcal{S}|}} \leq \theta \|g(x_k,|\mathcal{S}|)\|_2\right\}, \tag{5}$$

where $\mathcal{S}$ is the set of (gradient) observations sampled, the constant $\theta \in (0,1)$, and $\hat{\mathrm{Var}}(g(x_k,|\mathcal{S}|))$ is the sample variance of the gradient estimate $g(x_k,|\mathcal{S}|)$ constructed from the set $\mathcal{S}$ of gradient samples. (The variance estimate $\hat{\mathrm{Var}}(g(x_k,|\mathcal{S}|))$ will differ depending on whether sampling is independent and identically distributed (iid) and whether the underlying population is assumed to be finite; we have hence deliberately chosen to not provide an explicit expression for $\hat{\mathrm{Var}}(g(x_k,|\mathcal{S}|))$.) To reiterate, the salient feature of Algorithm 3 is the sample sizing rule (5) devised explicitly to keep the sampling error in the gradient estimate and a measure of proximity to a critical point in a fixed proportion.

Algorithm 3 (Dynamic Sampling Gradient Algorithm)
Require: Initial iterate x_0, initial sample $\mathcal{S}_0$, and constant $\theta \in (0,1)$.
1: set $k = 0$
2: **repeat**
3: compute $d_k = g(x_k, |\mathcal{S}_k|)$
4: set step length α_k (using line search or as a fixed step)
5: set $k = k + 1$
6: choose $\mathcal{S}_k$ such that $|\mathcal{S}_k| = |\mathcal{S}_{k-1}|$
7: compute sample variance
8: **until** a convergence condition is satisfied

The convergence and work complexity results in Byrd et al. [18] correspond to an idealized version of Algorithm 3 that uses the simplied sampling rule

$$m_k = \lceil a^k \rceil \text{ for some } a > 1 \tag{6}$$

instead of the adaptive sampling rule in (5). The main convergence and complexity result in Byrd et al. [18] is as follows.

Theorem 5. *Suppose the function f is L-smooth and λ-strongly convex, and it attains its minimum $x^* = \arg\inf_{x\in R^n} f(x)$. Then, the sequence $\{x_k\}$ generated by Algorithm 3 implemented with the sample size rule in (6) satisfies the following.*

(1) $\mathbb{E}[f(x_k) - f(x^*)] \leq C\rho^k$ *for all* $k \geq 1$, *where* $\rho = \max\{1 - \lambda/4L, a^{-1}\} < 1$ *and* $C = \max\{f(x_0) - f(x^*), 2\sigma^2/\lambda\}$.

(2) *The total number of gradient evaluations needed to obtain an ϵ-optimal solution—that is, a solution x_k satisfying $f(x_k) - f(x^*) \leq \epsilon$—is* $\mathcal{O}\left(\frac{L}{\lambda\epsilon} \max\{f(x_0) - f(x^*), 2\sigma^2/\lambda\}\right)$.

Three aspects of Theorem 5 are important. First, the theorem is proved for smooth and strongly convex functions; the corresponding complexity will be higher without the guaranteed presence of strong convexity (Hashemi et al. [43]). Second, afforded by structural conditions on f, the result holds for all $k \geq 1$, unlike typical results that one tends to see in the literature on stochastic approximation (Kushner and Yin [53]). Third, Theorem 5 assumes that a xed sample size sequence $m_k = \lceil a^k \rceil$ for some $a > 1$ is in effect.

As noted earlier, Theorem 5 pertains to an idealized version of Algorithm 3 that uses a xed sample size sequence. For effective implementation, Byrd et al. [18] suggest using Algorithm 3 with the adaptive sampling rule in (5), but with the direction-nding step 3 of Algorithm 3 augmented with a Hessian through the Newton-CG method, and the step length α_k in step 4 of Algorithm 3 obtained using a line search that satises the Wolfe conditions (Nocedal and

Wright [74]). Two issues are noted as especially important when incorporating the Hessian into the direction-nding step of Algorithm 3. First, to avoid excessive sampling, the sample size used for constructing the Hessian is such that it is retained at a constant factor R of the sample size obtained from the sampling rule in (5). Second, the Hessian should not be computed explicitly; instead, using what is called the Hessian-free *conjugate gradient* method (Nocedal and Wright [74]), a Hessian vector product is formed and incorporated directly. The implementable version of Algorithm 3 is listed as algorithm 5.2 and algorithm 5.1 in Byrd et al. [18].

Along the same lines as the idealized variant of Algorithm 3, Friedlander and Schmidt [36] propose what is called an *incremental gradient* method. In Friedlander and Schmidt [36], the growth condition on the sample sizes m_k is specied indirectly through an appropriate condition on the expected rate at which the error in the gradient estimate decays to zero. For example, Friedlander and Schmidt [36] prove the following result on the behavior of SGD's iterates.

Theorem 6. *Suppose the objective function f in Problem (P) is L-smooth and λ-strongly convex. Let*

$$x_{k+1} = x_k - \alpha_k\, g(x_k, m_k), \qquad k = 0, 1, 2, \ldots, \tag{7}$$

where m_k is such that $B_k := \mathbb{E}[\|g(x_k, m_k) - \nabla f(x_k)\|^2]$ satisfies

$$\lim_{k\to\infty} \frac{B_{k+1}}{B_k} \le 1.$$

Then, for any $\epsilon > 0$,

$$\mathbb{E}[f(x_k) - f(x^*)] \le (1 - \lambda/L)^k [f(x_0) - f(x^*)] + \mathbb{O}(C_k), \tag{8}$$

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

A particular construction of the sequence $\{B_k\}$ (albeit one that depends on a few unknown constants) is also presented in Friedlander and M. Schmidt [36] to ensure that the upper bound in (8) converges at a linear rate. An implementable version of the recursion in (7) that incorporates a Hessian approximation via limited-memory BFGS (Nocedal and Wright [74]) and a step length obtained on the basis of a line search with the Armijo condition (Nocedal and Wright [74]) is recommended in Friedlander and Schmidt [36].

4.2. Iterate-Averaging Methods

To motivate iterate-averaging as a general idea, let us recall, as was noted in Section 3, that basic SGD is optimal for L-smooth, c-strongly convex functions f with the step size choice $\alpha_k = \theta/k$ whenever $\theta > (2c)^{-1}$. Hence, as lucidly illustrated in Nemirovskii et al. [68], if we execute SGD on the L-smooth, c-strongly convex function $f(x) = x^2/10$ with $\alpha_k = \theta/k$ and $\theta > (2c)^{-1} = 2.5$, the resulting iterations will exhibit the optimal complexity rate $\mathbb{O}(k^{-1})$. However, if $\theta = 1$, it can be shown using simple algebra that the resulting complexity rate deteriorates to $\mathbb{O}(k^{-0.2})$. In other words, shorter steps $\alpha_k = k^{-1}$ resulting from any mis-estimation of the strong convexity constant c cause significant degradation in performance compared with the optimal complexity $\mathbb{O}(k^{-1})$. Such degradation can become even more pronounced when the underlying function is not strongly convex, as demonstrated through another example in Nemirovskii et al. [68].

Our illustration of the behavior of basic SGD on the function $f(x) = x^2/10$ is meant to convey the idea that step sizes cannot be chosen "too short" if the optimal complexity is to be guaranteed. However, "long steps" have been known to make SGD's trajectory "more noisy" (Nemirovskii et al. [68]). Iterate averaging is a technique that is intended as a balance between these extremes. Loosely speaking, iterate averaging allows using long steps within the basic SGD iteration but

then averages the resulting iterates offline, to account for the increased noise in the iterates. The general structure for such iterate-averaged SGD (for first-order oracles) is

$$x_{k+1} = x_k - \alpha_k g(x_k, 1);$$
$$\tilde{x}_{k+1} = \frac{1}{k+1}\sum_{i=1}^{k+1} x_i \quad k = 0, 1, 2, \ldots. \tag{9}$$

Iterate averaging as an idea first seems to have appeared in Nemirovsky and Yudin [69] for a setting more general than Euclidean spaces but was developed further in Polyak and Juditsky [79] and in Nemirovskii et al. [68]. A precise complexity rate result for the iterates resulting from (9) is given in Polyak and Juditsky [79] and is stated here in a slightly modified form.

Theorem 7 (Polyak and Juditsky [79]). *Let the function f be L-smooth, twice differentiable, and c-strongly convex with a unique minimum attained at x^*. Let $B(x)$ denote the $p \times p$ matrix of second derivatives of f at $x \in \mathcal{D}$. Let the noise process $\epsilon_k = G(x_k) - \nabla f(x_k)$ satisfy $\mathbb{E}[\epsilon_k \,|\, \mathcal{F}_{k-1}] = 0$ and $\mathbb{E}[\|\epsilon_k\|^2 \,|\, \mathcal{F}_{k-1}] + \|\nabla f(x_k)\|^2 \le K_2(1 + \|x_{k-1}\|^2)$ almost surely for some $K_2 > 0$. Also, let $\mathbb{E}[\epsilon_k \epsilon_k^T \,|\, \mathcal{F}_{k-1}] \xrightarrow{p} S$, where S is a positive denite matrix. Let the step size sequence $\alpha_k = \alpha k^{-\beta}$, where $1/2 < \beta < 1$. Then $\tilde{x}_{k+1} \to x^*$ almost surely, and $\sqrt{k}(x_k - x^*) \xrightarrow{d} \mathrm{N}(0, V)$, where $V = B(x^*)^{-1} S (B(x^*)^{-1})^T$.*

The crucial point to note about Polyak and Juditsky's [79] iterate averaging is that the convergence rate characterized in Theorem 7 is the best possible in an information-theoretic sense (see Section 6). And this best rate, crucially dependent on the second derivative of the function f at the point x^*, is attained with no explicit estimation of the second-derivative (Hessian) matrix. Iterate averaging and dynamic/adaptive sampling methods are intimately connected and essentially do the same thing but in different ways. The question of when to start averaging in iterate-averaging methods is a question of great practical importance about which little is currently known.

As noted earlier, a popular and general SGD technique that averages iterates akin to (9) is what has been called mirror descent (Nemirovskii et al. [68], Nemirovsky and Yudin [69]). Apparently, mirror descent was introduced as a generalization of the gradient descent iteration for non-Euclidean spaces, where the x_k iterate and the gradient $\nabla f(x_k)$ may not be in the same space, thus rendering an iteration such as $x_k - \gamma \nabla f(x_k)$ meaningless. To understand mirror descent, let us restrict our brief discussion here to a compact convex set $X \subset \mathbb{R}^d$ equipped with an arbitrary norm $\|\cdot\|$, even though mirror descent is designed for non-Euclidean spaces. The idea of mirror descent is simple in principle. Because an iteration such as $x_k - \gamma \nabla f(x_k)$ is meaningless, mirror descent performs the SGD iteration on the dual space, inverts back to the primal space, and then projects to the original space X to ensure feasibility. To make this operation precise, mirror descent denes a mirror map $\Phi : \mathcal{D} \to \mathbb{R}$ that is strictly convex and differentiable, having gradient taking all possible values in $\mathbb{R}^d$ and diverging on the boundary of $\mathcal{D}$; that is, $\lim_{x \to \partial\mathcal{D}} \|\nabla \Phi(x)\| = \infty$. The domain $\mathcal{D}$ of the mirror map is such that the original set X is a subset of its closure. Several example mirror maps have been presented; see, for example, section 4.3 in Bubeck [17]. Also, the Bregman divergence

$$D_\Phi(x, y) = \Phi(y) - \Phi(x) - \nabla \Phi(x)^T (y - x)$$

is used to accomplish the projection

$$\Pi_X^\Phi(y) := \arg\min D_\Phi(x, y)$$

in the primal space X. Given an iterate x_k, mirror descent first obtains the image $\Phi(x_k)$ of x_k through the mirror map $\Phi(\cdot)$, then performs the gradient step $\Phi(x_k) - \alpha_k \nabla g(x_k, m)$ to obtain the image $\nabla \Phi(y_{k+1})$ in the dual space, and inverts $\nabla \Phi(y_{k+1})$ to obtain y_{k+1}, which is finally projected back into the primal space. Formally, the iteration is written as

$$\begin{aligned}\nabla\Phi\big(y_{k+1}\big) &= \nabla\Phi(x_k) - \alpha_k g(x_k, m),\\ x_{k+1} &\in \Pi_X^{\Phi}\big(y_{k+1}\big).\end{aligned} \tag{10}$$

The "returned solution" $\tilde{x}_i^k$ after k steps, obtained by "nonuniform averaging" of the iterates $x_i, x_{i+1}, \ldots, x_k$, is given as

$$\begin{aligned}\tilde{x}_i^k &= \frac{\sum_{t=i}^k \gamma_t x_t}{\sum_{t=i}^k \gamma_t};\\ \gamma_t &= \frac{\theta D_{\Phi,X}\sqrt{\alpha}}{M_*\sqrt{t}}, \quad t = 1, 2, \ldots, k,\end{aligned} \tag{11}$$

where $\theta > 0$ is a chosen constant, the diameter

$$D_{\Phi,X} = \sqrt{2} \sup_{x\in X^o, z\in X} \big(\Phi(z) - \Phi(x) - \nabla\Phi(z)^T(x - Z)\big)^{1/2},$$

the constant α is the strong-convexity parameter of the function Φ, and $M_* \geq \sup_{x\in X} \mathbb{E}\big[\|g(x, m)\|_*^2\big]$.

As is noted in Nemirovskii et al. [68], mirror descent results in a convergence rate that is robust with respect to misestimation of the strong convexity parameter of the function f. The main complexity result for mirror descent given in Nemirovskii et al. [68] is

$$\mathbb{E}\big[f(\tilde{x}_i^k) - f(x^*)\big] \leq \frac{D_{\Phi,X} M_*}{\sqrt{\alpha k}} \left[\frac{2}{\theta}\frac{k}{k-i+1} + \frac{\theta}{2}\sqrt{\frac{k}{i}}\right]. \tag{12}$$

It can be observed that the convergence rate as implied by (12) is not optimal for smooth functions. As we will see in Section 6, the rate implied by (12) is optimal for nonsmooth functions.

4.3. Gradient Aggregation Methods

As was noted in the introductory paragraphs of Section 4, dynamic and adaptive sampling, iterate averaging, and gradient averaging are ways to "manage" the variance in the iterates of SGD to keep the resulting squared bias and the variance sequences in (4) in lock-step. Whereas dynamic/adaptive sampling and iterate averaging are now reasonably well developed, gradient aggregation as a method has not been widely investigated in the problem context considered in this tutorial. To be more specific, there has recently been much reported success with methods such as the stochastic variance reduced gradient (Johnson and Zhang [50]), SAGA (Defazio et al. [28]), and stochastic average gradient (Blatt et al. [12]), which estimate gradients through strategic aggregation. However, each of these methods relies crucially on computing what has been called the *full gradient* or the exact gradient at selected steps. Because the context of the current tutorial assumes that the underlying population from which gradient estimates are sampled is infinite, no such full gradient calculation option is relevant. Nevertheless, aggregating gradients is an idea that is worthy of investigation and that is bound to be competitive with dynamic/adaptive sampling and iterate averaging if done carefully.

5. SGD with Regularization

Regularization plays a key factor in formulation of a wide range of optimization models arising from statistical learning, signal processing, and distributed optimization. Moreover, the design and analysis of many trending first-order methods in recent years have been tailored to utilize the structure of regularized problems. Motivated by the significance of the role of regularization in formulation as well as development of solution methods, the goal of this section is to provide an introduction to regularization and its applications and to review some of the

well-known gradient-type methods and their stochastic variants for solving regularized optimization problems.

5.1. An Introduction to Regularization

The idea of regularization finds its roots in addressing *ill-posed* problems. In general, mathematical problems can be classified as being either well-posed or ill-posed. Hadamard [40] characterizes this definition as follows. Consider the equation

$$Az = u \qquad \text{for all } z \in Z \text{ and } u \in U, \tag{13}$$

where $A : Z \to U$ is an operator, and Z and U are metric spaces. Equation (13) is called well-posed if (i) for each $u \in U$, the equation $Az = u$ has a unique solution; and (ii) the solution is stable (i.e., not sensitive) to small perturbations made to the right-hand side of Equation (13) (cf. Tikhonov et al. [94]). In the optimization regime, consider the following canonical problem:

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in X, \end{array} \tag{P_0}$$

where $X \subseteq \mathbb{R}^n$ is a nonempty set and $f : X \to \mathbb{R}$ is a function. When problem (P_0) is ill-posed, a well-known approach is the regularization technique, where the goal is to construct a well-posed problem of the form

$$\begin{array}{ll} \min & f(x) + \lambda r(x) \\ \text{s.t.} & x \in X, \end{array} \tag{$\mathrm{P}_{\lambda,r}$}$$

where $\lambda \geq 0$ is a scalar and is referred to as the regularization parameter, and $r : \mathbb{R}^n \to \mathbb{R}$ is a suitable function called the regularizer. The parameter λ provides a trade-off between the fidelity of the objective f and the metric r. Throughout, we denote the solution to the regularized problem by x^*_λ.

5.2. Types of Regularization

Next, we present an overview of some of the popular types of regularization.

i. *Tikhonov regularization:* The classical and yet popular regularization technique is named after Andrey Tikhonov, where the regularizer is given by $r(x) = \frac{1}{2}\|x\|_2^2$, and $\|.\|_2$ denotes the Euclidean norm. The motivation in this type of regularization arises from the need to stabilize the solution of ill-posed linear inverse problems. The family of solutions $\{x^*_\lambda\}$ is called the Tikhonov trajectory. The properties of the Tikhonov trajectory have been studied in the literature extensively. For example, in Facchinei and Pang [32] (cf. section 12.2), continuity and boundedness of the set $\{x^*_\lambda \mid \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$ are established where $0 < \lambda_{\min} < \lambda_{\max}$ are arbitrary scalars. When λ decays to zero, the term $\lim_{\lambda \to 0} x^*_\lambda$ may not always exist. However, when the optimal solution set of problem (P_0) is nonempty, f is convex, and the set X is closed and convex, the Tikhonov trajectory converges to the least ℓ_2-norm optimal solution of problem (P_0).

ii. ℓ_1 *regularization:* An alternative to Tikhonov regularization is ℓ_1 regularization where we set $r(x) = \|x\|_1$. This technique has been widely used in machine learning, statistics, signal processing, and compressed sensing (e.g., Kim et al. [52]) to improve sparsity of the solution in large-scale problems. For example, in lasso regression (Tibshirani [93]), ℓ_1 regularization of linear least-squares problems is considered.

iii. ℓ_∞ *regularization:* Another regularization scheme is the max-norm regularization where $r(x) = \|x\|_\infty$. This technique has been applied in collaborative filtering problems (e.g., Srebro et al. [93]).

iv. *Frobenius norm regularization:* The regularization technique is also applied in optimization problems on matrix spaces. A common form is the Frobenius norm regularization where

$r(X) = \|X\|_F^2$. An example is in matrix completion problems where the goal is estimating a low-rank matrix given a linear system of equations (Recht et al. [80], Wright [98]).

5.3. Applications in Linear Inverse Problems

The class of inverse problems (in contrast with "direct problems") has a rich and well-studied literature. Starting in the 1980s, the past few decades have seen much interest in the theory and applications of this class of problems. In particular, a diverse range of applications arising from image reconstruction, signal processing, astrophysics, and statistical learning can be cast as linear inverse problems (Beck and Teboulle [7]). In these models, a discrete linear system of the following form is considered:

$$Ax = b + \delta, \tag{14}$$

where given $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$; and $\delta \in \mathbb{R}^m$ is an unknown noise. We seek to compute a vector $x \in \mathbb{R}^n$ that represents the true value of an image or a received signal. To motivate the relevance of ill-posedness to the class of linear inverse problems, consider the following toy example. Let A and b be given by

$$A = \begin{bmatrix} 0.835 & 0.667 \\ 0.333 & 0.266 \end{bmatrix}, \qquad b = \begin{bmatrix} 0.168 \\ 0.067 \end{bmatrix}.$$

The exact (true) solution to the system $Ax = b$ is $x^* = [1\ -1]^T$. Now consider the case where b is perturbed by a small noise $\delta = [0\ -10^{-3}]^T$. The exact solution changes dramatically to $\hat{x} = [-666\,834]^T$. This is because matrix A is ill-conditioned, meaning that it has a relatively large condition number (Meyer [66]). This is indeed the case in a wide range of applications as well.

Next, we illustrate the challenges of ill-posedness in an image deblurring application and show how regularization can be used as a remedy for ill-posed problems. Consider the linear inverse problem (14) where $m = n$. Given a blurred image $b \in \mathbb{R}^n$ and a blur operator $A \in \mathbb{R}^{n\times n}$, the goal is to find an unknown (true) image $x \in \mathbb{R}^n$. A classical approach is to solve the least-squares problem of the form

$$\hat{x} \in \arg\min_{x\in\mathbb{R}^n} \|Ax - b\|_2^2. \tag{LS}$$

Often in image deblurring, the blur operator A is ill-conditioned. Consequently, solving the least-squares problem may result in a noisy, and sometimes meaningless, solution. This issue can be addressed using Tikhonov regularization where the following problem is considered:

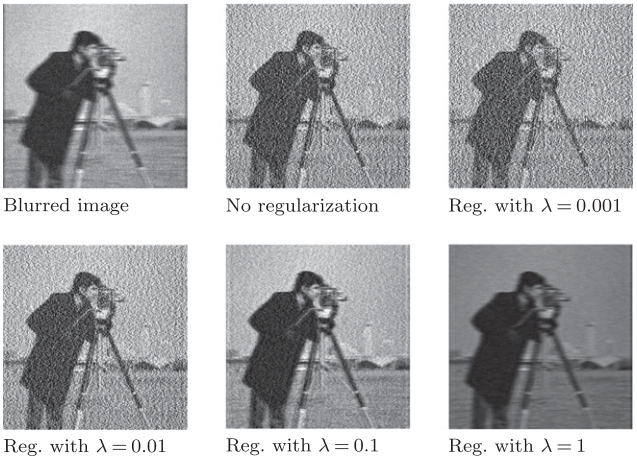
$$\hat{x} \in \arg\min_{x\in\mathbb{R}^n} \|Ax - b\|_2^2 + \frac{\lambda}{2}\|x\|_2^2. \tag{LS$_\lambda$}$$

Figure 1 illustrates this in an example where we employ the standard deterministic gradient method for solving (LS$_\lambda$). When regularization is suppressed (i.e., when problem (LS) is solved), the deblurred image is perturbed with noises. This is also the case when the regularization parameter is relatively small (i.e., $\lambda = 0.001$ and 0.01 in this case). When λ is relatively too large (i.e., 1 in this case), the picture becomes blurry again. Using a moderate choice of $\lambda = 0.1$, a relatively better deblurred image is generated. This motivates the following discussion on the subject of *exact regularization*.

5.4. Exact Regularization

A natural question in employing the regularization technique pertains to the choice of the regularization parameter. In general, when $\lambda > 0$, the solution(s) to the regularized problem (P$_{\lambda,r}$) may not solve the original problem (P$_0$). Although this deviation is often acceptable

Figure 1. Impact of Tikhonov regularization in image deblurring.



given the benefits of the well-posed regularized problem, it would be more desirable if the regularization is exact (i.e., the solutions to problem $(\mathrm{P}_{\lambda,r})$ also solve the original problem (P_0)). The necessary and sufficient conditions to ascertain exact regularization were studied by Bertsekas [8], Bertsekas et al. [9], Mangasarian and Meyerf [63], Ferris and Mangasarian [34], and Friedlander and Tseng [37], and more recently by Charitha et al. [21] addressing ill-posed variational inequalities. Especially in the optimization regime, exact regularization is tied closely with the following "selection problem":

$$\begin{aligned} &\min\ r(x) \\ &\text{s.t.}\quad x \in X^*, \end{aligned} \tag{P_r}$$

where X^* denotes the optimal solution set of problem (P_0). In Friedlander and Tseng [37], it is shown that the regularization $(\mathrm{P}_{\lambda,r})$ is exact if and only if the selection problem (P_r) has a Lagrange multiplier μ^*. Moreover, for any $\lambda < 1/\mu^*$, we have $X^*_\lambda = X^*_r$, where $X^*{}_\lambda$ and X^*_r denote the optimal solution sets of problems $(\mathrm{P}_{\lambda,r})$ and (P_r), respectively. This is formally presented in the following.

Theorem 8 (Theorem 2.1 in Friedlander and Tseng [37]). *Consider problems* (P_0), $(\mathrm{P}_{\lambda,r})$, *and* (P_r). *Let the feasible set* X *and the optimal solution set* X^* *be nonempty. Suppose the level set* $\{x \in X^* \mid r(x) \le \beta\}$ *is bounded for all* $\beta \in \mathbb{R}$, *and* $\inf_{x\in X} r(x) > -\infty$ *(e.g., this holds when* r *is coercive).*

(a) *For any* $\lambda > 0$, $X^* \cap X^*_\lambda \subseteq X^*_r$.
(b) *If there exists a Lagrange multiplier* μ^* *for* (P_r), *then* $X^* \cap X^*_\lambda = X^*_r$ *for all* $\lambda \in (0, 1/\mu^*]$.
(c) *If there exists* $\bar{\lambda} > 0$ *such that* $X^* \cap X^*_{\bar{\lambda}} \neq \emptyset$, *then* $1/\bar{\lambda}$ *is a Lagrange multiplier for* (P_r), *and* $X^* \cap X^*_\lambda = X^*_r$ *for all* $\lambda \in (0, \bar{\lambda}]$.
(d) *If there exists* $\bar{\lambda} > 0$ *such that* $X^* \cap X^*_{\bar{\lambda}} \neq \emptyset$, *then* $X^*_\lambda \subseteq X^*$ *for all* $\lambda \in (0, \bar{\lambda})$.

When the regularization is not exact, Friedlander and Tseng [37] derive error bounds on the distance from the regularized solution to the original solution set (see theorem 5.1 in Friedlander and Tseng [37]).

The remainder of this section is focused on the solution methods for solving regularized problems. We review some of the popular rst-order methods including SGD, proximal gradient methods and their stochastic variants, and block proximal gradient methods for solving regularized problems of the form $(\mathrm{P}_{\lambda,r})$.

5.5. Standard SGD for ℓ_2-Regularized Problems

In this section, we review the application of SGD in solving regularized problems of the form $(P_{\lambda,r})$ in the stochastic regime. To this end, motivated by applications of ℓ_2 regularization in statistical learning (e.g., support vector machines), we first consider the following problem:

$$\min\ f_\lambda^{Tikh}(x) \triangleq \mathbb{E}[F(x,\xi)] + \frac{\lambda}{2}\|x\|_2^2$$
$$\text{s.t. } x \in X,$$

where $F : \mathbb{R}^n \times \Omega \to \mathbb{R}$ is a stochastic function, the random vector ξ is given as $\xi : \Omega \to \mathbb{R}^d$, $(\Omega, \mathcal{F}, \mathbb{P})$ denotes the associated probability space, and the expectation $\mathbb{E}[F(x,\xi)]$ is taken with respect to $\mathbb{P}$. We consider the following assumption.

Assumption 1. *Let the following hold:*

(a) *The set $X \subseteq \text{int}(\text{dom}(f))$ is nonempty, convex, and closed.*
(b) *The function $f(x) \triangleq \mathbb{E}[F(x,\xi)]$ is convex over X.*

The outline of the algorithm is provided by Algorithm 4. We let $\mathcal{P}_X(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ denote the projection operator onto set X, and let $\mathcal{F}_k$ denote $\{x_0, \xi_0, \xi_1, \ldots, \xi_{k-1}\}$. The convergence analysis will be discussed for smooth and nonsmooth cases according to the following assumption.

Assumption 2 (Nonsmooth Case). *Let the subgradient $G(x,\xi) \in \partial F(x,\xi)$ satisfy the following:*

(a) *For all $x \in X$, we have $\mathbb{E}[G(x,\xi) \mid x] \in \partial f(x)$.*
(b) *There exists a scalar $C > 0$ such that $\mathbb{E}[\|G(x,\xi)\|_2^2] \le C^2$ for all $x \in X$.*

Assumption 3 (Smooth Case). *Let $F(x,\xi)$ be differentiable for all $\xi \in \Omega$, let $G(x,\xi)$ denote the gradient map of F, and suppose the following hold:*

(a) *For all $x \in X$, we have $\mathbb{E}[G(x,\xi) \mid x] = \nabla f(x)$.*
(b) *There exists a scalar $\sigma > 0$ such that $\mathbb{E}[\|G(x,\xi) - \nabla f(x)\|_2^2] \le \sigma^2$ for all $x \in X$.*
(c) *Function f is L_f-smooth; that is, ∇f is Lipschitz with parameter L_f.*

The following result characterizes the error of Algorithm 4 in terms of a recursive bound in smooth and nonsmooth cases (cf. Yousean et al. [104] and [62]).

Lemma 1 (Recursive Error Bounds for Smooth and Nonsmooth Cases). *Consider problem (5.5). Let Assumption 1 hold, and let the sequence $\{x_k\}$ be generated by Algorithm (4) using an arbitrary step size sequence $\{\alpha_k\}$.*

(a) *If Assumption 2 holds, then*
$$\mathbb{E}\big[\|x_{k+1} - x^*\|_2^2 \mid \mathcal{F}_k\big] \le (1 - 2\lambda\alpha_k)\|x_k - x^*\|_2^2 + C^2\alpha_k^2.$$

(b) *If Assumption 3 holds, then*
$$\mathbb{E}\big[\|x_{k+1} - x^*\|_2^2 \mid \mathcal{F}_k\big] \le \big(1 - 2\lambda\alpha_k + 2\big(L_f + \lambda\big)^2\alpha_k^2\big)\|x_k - x^*\|_2^2 + 2\sigma^2\alpha_k^2.$$

The convergence analysis and complexity result of Algorithm 4 are provided as follows.

Algorithm 4 (Self-Tuned Stochastic (Sub)Gradient Algorithm; Youseffan et al. [104])

Require: Initial point x_0 and initial step size α_0
1: set $k = 0$
2: **repeat**

3: set $x_{k+1} = \mathcal{P}_X(x_k - \alpha_k(G(x_k, \xi_k) + \lambda x_k))$
4: set $\alpha_{k+1} = \alpha_k(1 - \lambda\alpha_k)$
5: set $k = k + 1$
6: **until** a convergence condition is satisfied

Theorem 9 (Youseffan et al. [104]). *Consider problem* (5.5). *Let Assumption* 1 *hold, and let* $\{x_k\}$ *be generated by Algorithm* 4. *Also, let either Assumption* 2 *or* 3 *hold. Suppose the set* X *is bounded; that is, there is* $\theta_X > 0$ *such that* $\|x\|_2 \le \theta_X$ *for all* $x \in X$.

(a) *Then the sequence* $\{x_k\}$ *converges almost surely to the unique optimal solution* x^* *of problem* (5.5).

(b) *There exists a threshold* $0 < \bar{\alpha} \le \frac{1}{2\lambda}$ *such that for any* $\alpha_0 \le \bar{\alpha}$ *and any* $k \ge 1$, *the self-tuned step sizes* $(\alpha_0, \alpha_1, \ldots, \alpha_{k-1})$ *given by Algorithm* 4 *minimize a mean squared error given by Lemma* 1 *for all* $(\alpha_0, \alpha_1, \ldots, \alpha_{k-1}) \in \left(0, \frac{1}{2\lambda}\right]^k$.

(c) *In the nonsmooth case, if Assumption* 2 *holds with* $C \ge \sqrt{8}\theta_X\lambda$, *then let* $\alpha_0 = \frac{4\lambda\theta_X^2}{C^2}$. *We have*

$$\mathbb{E}\left[\|x_k - x^*\|_2^2\right] \le \left(\frac{C}{\lambda}\right)^2 \frac{1}{k} \qquad \text{for all } k \ge 1.$$

Moreover, let ϵ *and* ρ *be arbitrary positive scalars, and let* $K \triangleq \left(\frac{3C^2}{\lambda^2}\right)\frac{1}{\epsilon\rho}$. *We have for all* $k \ge K$

$$\text{Prob}\left(\|x_j - x^*\|_2^2 \le \epsilon \text{ for all } j \ge k\right) \ge 1 - \rho.$$

(d) *In the smooth case, if Assumption* 3 *holds, then let* $\alpha_0 = \frac{2\lambda\theta_X^2}{\sigma^2 + 4\theta_X^2(L_f+\lambda)^2}$. *We have*

$$\mathbb{E}\left[\|x_k - x^*\|_2^2\right] \le 2\left(\frac{\sigma^2 + 4\theta_X^2(L_f+\lambda)^2}{\lambda^2}\right)\frac{1}{k} \qquad \text{for all } k \ge 1.$$

Moreover, let ϵ *and* ρ *be arbitrary positive scalars, and let* $K \triangleq 6\left(\frac{\sigma^2+4\theta_X^2(L_f+\lambda)^2}{\lambda^2}\right)\frac{1}{\epsilon\rho}$. *We have for all* $k \ge K$

$$\text{Prob}\left(\|x_j - x^*\|_2^2 \le \epsilon \text{ for all } j \ge k\right) \ge 1 - \rho.$$

This result addresses the convergence of SGD for ℓ_2-regularized problem. Next, we review some of the popular methods for solving more generalized variants of regularized problems.

5.6. Proximal Gradient Methods and Their Stochastic Variants

To induce sparsity, ℓ_1 regularization has been extensively applied in the literature. Note that despite the ℓ_2 regularizer, the ℓ_1-norm is neither strongly convex nor smooth. Although strong convexity and smoothness often help with acceleration of the first-order methods, the lack of these properties for the regularizer introduces challenges in the design of fast optimization algorithms. Particularly, when the main objective function is smooth and is regularized using a nonsmooth regularizer such as the ℓ_1-norm, a natural question is if one may develop an "optimal" first-order method to solve the regularized problem. In recent years, this question has led to the evolution of an important class of optimization schemes called proximal methods. In this section, we review these methods and provide some examples. Consider the following optimization model:

$$\begin{aligned} &\min\ f(x) + g(x) \\ &\text{s.t.}\ \ x \in \mathbb{R}^n, \end{aligned} \tag{15}$$

where f is a proper, convex, and smooth function, and g is convex but possibly nondifferentiable. Problem (15) is referred to as *composite optimization*. A wide range of problems can be cast as a composite model. Among these include the following two important classes of problems:

i. *Constrained optimization*: When $g(x)$ is assumed to be the indicator function of a convex set $X \subseteq \mathbb{R}^n$, problem (15) is equivalent to the constrained minimization $\min_X f(x)$.

ii. *Nonsmooth regularization of smooth functions*: For example, when $g(x) = \lambda\|x\|_1$, model (15) captures ℓ_1 regularization.

Proximal first-order methods are appealing in the sense that they utilize the structure of the composite model to recover fast convergence rates of smooth optimization. For example, in solving (15), while employing the standard subgradient method, the convergence rate is of the order $1/\sqrt{k}$; employing an accelerated proximal-gradient method, the rate improves to $1/k^2$. At each iteration of the proximal methods, a proximal mapping (also referred to as the prox mapping) is computed that is defined as follows.

Definition 1 (Proximal Mapping). Given a function $h : \mathbb{R}^n \to (-\infty, +\infty]$, the proximal mapping of h is the operator given by

$$\text{prox}_h(x) \triangleq \arg\min_{u \in \mathbb{R}^n} \left\{ h(u) + \frac{1}{2}\|u - x\|_2^2 \right\}.$$

An extensive review of properties of the prox operator is provided in Beck [6] (see chaps. 6 and 10). Of these, we note that the prox mapping is a generalization of the projection operator in the sense that $\text{prox}_{\delta_X}(x) = \mathcal{P}_X(x)$, where $\delta_X(\cdot)$ denotes the indicator function associated with the set X, and $\mathcal{P}_X(\cdot)$ denotes the projection operator onto the set X. Moreover, in ℓ_1-regularized problems where $g(x) = \lambda\|x\|_1$, we have $\text{prox}_{\lambda\|\cdot\|_1}(x) = \mathcal{T}_\lambda(x)$, where $\mathcal{T}_\lambda(x)$ is the soft thresholding mapping defined as follows.

Definition 2 (Soft Thresholding Mapping). Given a scalar λ, the soft thresholding mapping is given by

$$\mathcal{T}_\lambda(x) \triangleq \text{sign}(x) \odot [\,|x| - \lambda \mathbf{1}_n]_+ \qquad \text{for all } x \in \mathbb{R}^n,$$

where $[u]_+ \triangleq \max\{\mathbf{0}_n, u\}$ for some $u \in \mathbb{R}^n$, and $\odot$ denotes the componentwise product of two vectors.

Intuitively speaking, given a vector x, $\mathcal{T}_\lambda(x)$ generates a sparse approximation of x that is characterized in terms of a parameter $\lambda > 0$. Motivated by this property, the soft thresholding mapping has played an important role in development of optimization methods in the literature addressing ℓ_1-regularized problems. One of the most popular variants of proximal methods is the class of iterative shrinkage-thresholding algorithms (ISTA) for solving problem (15), where $f(x) = \|Ax - b\|_2^2$ and $g(x) = \lambda\|x\|_1$ (see, e.g., Chambolle et al. [20], Daubechies et al. [27], Figueiredo and Nowak [35], Hale et al. [41], and Wright et al. [100], and references in Beck and Teboulle [7]). At each iteration of ISTA, the update rule is as follows:

$$x_{k+1} = \mathcal{T}_{\lambda\alpha}\big(x_k - 2\alpha A^T(Ax_k - b)\big),$$

where $\alpha > 0$ is an appropriate step size. ISTA attains the worst-case complexity result of the order $1/k$. Similar methods to ISTA have been proposed and studied under different names. Examples include the truncated gradient in Langford et al. [55] and forward-backward splitting methods that have been increasingly applied in signal processing (Combettes and Pesquet [25], Duchi and Singer [30], Lions and Mercier [59], Rosasco et al. [83]). An accelerated variant of ISTA for solving problem (15), referred to as the fast iterative shrinkage-thresholding algorithm (FISTA), was developed in Beck and Teboulle [7] and achieves the rate of the order $1/k^2$. A variant of FISTA when the Lipschitzian parameter of ∇f is known is presented in Algorithm 5.

Algorithm 5 (FISTA with Constant Step Size)

Require: Initial point x_0 and Lipschitzian parameter of ∇f
1: set $k = 0$, $t = 1$, and $y_1 = x_0$
2: **repeat**
3: set $x_k = \text{prox}_{\frac{1}{L}g}\left(x_k - \frac{1}{L}\nabla f(x_k)\right)$
4: set $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
5: set $y_{k+1} = x_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(x_k - x_{k-1})$
6: set $k = k + 1$
7: **until** a convergence condition is satisfied

When the computation of the proximal mapping is corrupted by an error, the topic of *inexact proximal gradient methods*, studied in Schmidt et al. [85], comes up. It is shown that if the error in the computation of the proximal mapping is controlled in an appropriate way, inexact proximal gradient schemes achieve the same convergence rates as the corresponding exact variants.

In the stochastic regime where the function f is given as an expected value, proximal stochastic gradient and mirror descent schemes have been developed more recently. Consider the following ℓ_1-regularized stochastic optimization problem:

$$\begin{aligned} &\min\ f_\lambda^{\ell_1}(x) \triangleq \mathbb{E}[F(x,\xi)] + \lambda\|x\|_1 \\ &\text{s.t.}\quad x \in \mathbb{R}^n. \end{aligned} \tag{16}$$

In solving problem (6), a proximal first-order method called the stochastic mirror descent made sparse was developed in Shalev-Shwartz and Tewari [86], extending the truncated gradient method in Langford et al. [55] to non-Euclidean settings. In Euclidean settings, a more generalized variant of this method is provided in Algorithm 6.

Algorithm 6 (Stochastic Gradient Method Made Sparse)

Require: Initial point x_0 and step size sequence α_k
1: set $k = 0$
2: **repeat**
3: set $x_{k+1} = \mathcal{T}_{\lambda\alpha_k}(x_k - \alpha_k G(x_k, \xi_k))$
4: set $k = k + 1$
5: **until** a convergence condition is satisfied

The following result provides the convergence analysis of Algorithm 6 under constant step sizes (Shalev-Shwartz and Tewari [86]).

Theorem 10. *Consider problem (16). Let Assumptions 1 and 2 hold, and let $\{x_k\}$ be generated by Algorithm 6. Set $x_0 = 0$ and $\alpha_k = \alpha$ for some scalar $\alpha > 0$ for all k. Let $K \geq 1$ be a fixed integer, and let t be selected from $\{0, 1, \ldots, K-1\}$ using a uniform discrete distribution. Then, we have*

$$\mathbb{E}\left[f_\lambda^{\ell_1}(x_t)\right] - f_\lambda^{\ell_1}(x^*) \leq \frac{\alpha C^2}{2} + \frac{\|x^*\|_1^2}{\alpha K},$$

where x^* is an optimal solution of problem (16). When $\alpha = \frac{2}{C\sqrt{K}}$, we have

$$\mathbb{E}\left[f_\lambda^{\ell_1}(x_t)\right] - f_\lambda^{\ell_1}(x^*) \leq \left(1 + \|x^*\|_1^2\right)\frac{C}{\sqrt{K}}.$$

More recently, extensions of Algorithm 6 for addressing the case that g is a general nonsmooth function have been studied in Rosasco et al. [83]. In the next section, we review generalizations of proximal methods to solve huge-scale optimization problems in deterministic and stochastic regimes.

5.7. Block Proximal Gradient Methods

A common challenge in solving optimization problems, particularly applications that arise from big data, is huge dimensionality of the solution space. Generally speaking, for an optimization problem in $\mathbb{R}^n$, when n exceeds 10^{12}, the direct implementation of the first-order methods becomes problematic. This is because, for example, standard SGD at each iteration requires performing arithmetic operations of order n. To address this issue and improve the efficiency of the underlying solution method, coordinate descent (CD) and, more generally, block coordinate descent (BCD) methods have been developed and studied in recent decades. Ortega and Rheinboldt [75] study the concept of such approaches as "univariate relaxation." The convergence properties of the CD methods were studied in the 1980s and the 1990s by researchers such as Tseng (Luo and Tseng [60, 61], Tseng and Yun [95]), Bertsekas, and Tsitsiklis (Bertsekas and Tsitsiklis [9]), and more recently in Beck [5], Mareček et al. [64], Richtárik and Takáč [81], Wright [98], and Xu and Yin [101] (see Wright [99] for a detailed review of CD methods). The key idea in standard block coordinate schemes is that at each iteration of the underlying solution method, only a block of the generated sequence(s) is updated requiring computation of only one block of the gradient mapping(s). There are different policies for choosing blocks in BCD schemes. Of these, in *cyclic* schemes, a block is chosen successively in a cyclic (deterministic) manner. Another popular strategy is *randomized* block coordinate schemes, where at each iteration, a randomly chosen block of the generated sequence(s) is updated.

In this section, we review some of the major contributions in the area of block coordinate methods and their proximal extensions for solving large-scale composite optimization problems. Throughout, we use the following notation. Let $n \triangleq \sum_{i=1}^{b} n_i$, where n_i is the dimension of the ith block and b is the number of blocks. For any vector $x \in \mathbb{R}^n$, let $x^i \in \mathbb{R}^{n_i}$ denote the ith block coordinate of x such that $x = (x^1; x^2; \ldots; x^b)$. We use subscript i to denote the ith block of the gradient mapping in $\mathbb{R}^n$ (e.g., we have $\nabla f(x) = (\nabla_1 f(x); \nabla_2 f(x); \ldots; \nabla_b f(x))$).

Nesterov [72] appears to be the first to develop nonasymptotic convergence rates for randomized block coordinate methods without imposing any strong assumptions. In the following, we briefly review some of the main contributions in Nesterov [72].

Consider the unconstrained minimization problem $\min_{x\in\mathbb{R}^n} f(x)$, where f is convex, and the gradient of function f is blockwise Lipschitz continuous with parameters L_i. The Euclidean variant of randomized coordinate descent method in Nesterov [72] is given by Algorithm 7. In Nesterov [72], convergence rates in terms of the error $\mathbb{E}[f(x_k)] - f^*$ are derived for problems with convex and strongly convex objectives, where the expectation is with respect to $\{i_j\}_{j=0}^{k-1}$. When f is a convex function, a sublinear rate of convergence is derived, whereas for a strongly convex objective function, the error bound improves to a linear rate characterized by the strong convexity parameter. In the unconstrained case, an accelerated variant of this scheme, called ACDM, was also developed in Nesterov [72], where a rate of the order $\frac{1}{k^2}$ was derived under strong convexity. Probabilistic bounds characterizing the convergence rate of the developed methods were also provided in Nesterov [72]. Under a uniform distribution for choosing block coordinates, the expected complexity of this scheme for finding an ϵ-solution for unconstrained minimization problems is $\mathcal{O}\left(\frac{b}{\sqrt{\epsilon}} \max_{1\le i\le b} L_i\right)$. This complexity bound was improved in Lee and Sidford [56] to $\mathcal{O}\left(\sqrt{\frac{b}{\epsilon}\sum_{i=1}^{b} L_i}\right)$ by employing the probabilities for choosing the coordinates as $L_i\left(\sum_{j=1}^{b} L_j\right)^{-1}$. Recently, Nesterov and Stich in [73] have derived a new complexity bound for a variant of the ACDM scheme of Nesterov [72]. The authors suggest using the probabilities $\sqrt{L_i}\left(\sum_{j=1}^{b}\sqrt{L_j}\right)$ and derive the complexity

$$\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\sum_{i=1}^{b}\sqrt{L_i}\right).$$

Algorithm 7 (Randomized Coordinate Descent Method)

Require: Initial point x_0, block Lipschitzian parameters L_i, and probabilities $\{p_i\}_{i=1}^b$

1: set $k = 0$

2: **repeat**

3: generate a random variable i_k such that $\text{Prob}(i_k = i) = p_i$

4: for $i \in \{1, \ldots, b\}$, set $x_{k+1}^i = \begin{cases} x_k^i - \frac{1}{L_i}\nabla f_i(x_k) & \text{if } i = i_k, \\ x_k^i & \text{if } i \neq i_k, \end{cases}$

5: set $k = k + 1$

6: **until** a convergence condition is satisfied

The signicance of this result is that for the rst time, the complexity estimate of the proposed ACDM method does not depend explicitly on the dimension of the space of variables. It is shown that this method often outperforms the standard accelerated gradient method.

Before we proceed with the proximal extensions of the block coordinate schemes, we review a randomized block coordinate scheme by Dang and Lan [26] for solving constrained stochastic optimization problems. Consider the following block-structured nonsmooth stochastic optimization problem:

$$\begin{aligned} \min \quad & f(x) \triangleq \mathbb{E}[F(x, \xi)] \\ \text{s.t.} \quad & x \in X \triangleq \prod_{i=1}^{b} X_i, \end{aligned} \tag{SCOP}$$

where $\xi \in \mathbb{R}^d$ is a random variable associated with a probability distribution, the function $F(\cdot, \xi) : X \to \mathbb{R}$ is continuous for all ξ, and the set $X \in \mathbb{R}^n$ is the Cartesian product of the sets $X_i \in \mathbb{R}^{n_i}$, with $n \triangleq \sum_{i=1}^b n_i$. A stochastic block mirror descent (SBMD) method for solving large-scale nonsmooth and stochastic optimization problems is developed in Dang and Lan [26]. An equivalent representation of the SBMD method in a Euclidean setting is provided in Algorithm 8. Here, $G_i(x, \xi)$ denotes the ith block of $G(x, \xi)$, where $G(x, \xi) \in \partial F(x, \xi)$, and $\mathcal{P}_{X_i}(\cdot)$ denotes the projection operator onto the set X_i. The complexity analysis of the SBMD method for both convex and strongly convex cases was studied in Dang and Lan [26]. In the case where the objective f is convex, the expected convergence rate is of the order $\sqrt{\frac{b}{k}}$, whereas in the case where f is strongly convex and nonsmooth, the expected rate improves to the order $\frac{b}{\mu k}$, where μ is the strong convexity parameter. Note that Algorithm 8 addresses general stochastic nonsmooth problems, and consequently, it can be applied for addressing regularized stochastic optimization problems.

Algorithm 8 (Randomized Block SGD)

Require: Initial point $x_0 \in X$, a step size sequence $\{\gamma_k\}_{k \geq 0}$, probabilities $\{p_i\}_{i=1}^b$, and $S_0 = 0$

1: Set $k = 0$

2: **repeat**

3: for $i \in \{1, \ldots, b\}$, set $x_{k+1}^i = \begin{cases} \mathcal{P}_{X_i}(x_k^i - \gamma_k G_i(x_k, \xi_k)) & \text{if } i = i_k, \\ x_k^i & \text{if } i \neq i_k, \end{cases}$

4: update S_k and $\bar{x}_k$ using the following recursions

$$S_{k+1} = S_k + \gamma_k$$
$$\bar{x}_{k+1} = \frac{S_k \bar{x}_k + \gamma_k x_{k+1}}{S_{k+1}}$$

5: set $k = k + 1$

6: **until** a convergence condition is satisfied

Recently, Yousean et al. [105] have developed randomized block coordinate stochastic extragradient schemes to solve stochastic Cartesian variational inequalities and derived similar rates as in Dang and Lan [26] under weaker assumptions using nonaveraging schemes.

Nesterov [72] also developed randomized coordinate schemes for constrained optimization that were later extended by Richtárik and Takáč in [81] to proximal schemes for solving composite optimization problems. Next, we review a randomized coordinate descent for composite functions named uniform coordinate descent for composite (UCDC) functions that was developed in Richtárik and Takáč [81]. The problem of interest is the following composite model:

$$\begin{array}{ll} \min & F(x) \triangleq f(x) + \sum_{i=1}^{b} g_i(x^i) \\ \text{s.t.} & x \in \mathbb{R}^n, \end{array} \tag{17}$$

where f is a smooth convex function, and $g(x) \triangleq \sum_{i=1}^{b} g_i(x^i)$ is a simple nonsmooth block-separable convex function. Let us define a block proximal mapping T_α^i as follows:

$$T_\alpha^i(x) \triangleq \operatorname{prox}_{\frac{1}{\alpha} g_i}\left(x_i - \frac{1}{\alpha}\nabla f_i(x)\right), \tag{18}$$

where $\alpha > 0$ is a scalar. An equivalent representation of UCDC is provided by Algorithm 10. In the case that f is convex, theorem 5 in Richtárik and Takáč [81] states that the sequence generated by the UCDC method obtains an ϵ-accurate solution with probability at least $1-\rho$ in at most $\mathcal{O}((b/\epsilon)\log(1/\rho))$ iterations. In this case, a sublinear convergence rate can also be derived for $\mathbb{E}[F(x_k)] - F^*$ where F is given by (17). When f is strongly convex, the former complexity bound improves to $\mathcal{O}(b\log(1/(\epsilon\rho)))$, whereas the latter bound improves to a linear rate (cf. theorems 7 and 8 in Richtárik and Takáč [81]).

Algorithm 9 (Uniform Coordinate Descent for Composite Functions)

Require: Initial point x_0 and block Lipschitzian parameters L_i

1: Set $k = 0$

2: **repeat**

3: Generate a uniform random variable $i_k \in \{1, \ldots, b\}$,

4: for all $j \in \{1, \ldots, b\}$, set $x_{k+1}^i = \begin{cases} T_{L_i}^i(x_k) & \text{if } i = i_k, \\ x_k^i & \text{if } i \neq i_k, \end{cases}$

5: **until** a convergence condition is satisfied

Accelerated variants of the UCDC method were developed in Fercoq and Richtárik [33] and Lin et al. [58]. These methods are named APPROX (accelerated, parallel, and proximal) and APCG (accelerated randomized proximal coordinate gradient), respectively. The APPROX method achieves an accelerated sublinear convergence rate. But it cannot exploit the strong convexity of the objective function to obtain accelerated linear rates in the composite case. This limitation is addressed by APCG. The APCG method achieves accelerated linear convergence rates when the composite objective function is strongly convex. Without the strong convexity assumption, APCG recovers a special case of the APPROX method.

Next, we briefly review cyclic block coordinate schemes. A more comprehensive review of these schemes can be found in Beck [6]. Consider the composite model (17). Luo and Tseng [60] develop a cyclic proximal gradient method where the nonsmooth functions g_i are indicator functions. They show that under the assumptions of (i) strong convexity with respect to each block, (ii) the existence of a local error bound on the objective function, and (iii) proper separation of isocost surfaces, a linear rate of convergence can be derived. Extending the results of Luo and Tseng [60], Beck and Tetruashvili [4] develop a cyclic block coordinate gradient projection method called BCGD. A sublinear rate of convergence is established, and it is shown that it can be accelerated for unconstrained problems. Generalizations of BCGD for addressing the general composite model (17) were studied by She and Teboulle [92] and more recently by Hond et al. [49]. Below, we present the standard cyclic block proximal gradient (CBPG) method for solving problem (17). Under convexity assumption on f, the CBPG method

achieves a sublinear convergence rate in terms of the objective value F given by (17) (cf. chapter 11 in Beck [6]).

Algorithm 10 (Cyclic Block Coordinate Proximal Gradient Method)

Require: Initial point x_0 and block Lipschitzian parameters L_i
1: Set $k = 0$ and $y_{k,0} = x_0$
2: **repeat**
3: **for** $i \in \{1, \ldots, b\}$ **do**
4: for all $j \in \{1, \ldots, b\}$, set $y^j_{k,i} = \begin{cases} T^i_{L_i}(y_{k,i-1}) & \text{if } j = i, \\ y^j_{k,i-1} & \text{if } j \neq i, \end{cases}$
5: **end for**
6: Set $x_{k+1} = y_{k,b}$
7: **until** a convergence condition is satisfied

6. Optimal Methods

By *optimal methods*, we mean SGD variants that achieve the fastest achievable convergence rate. More precisely, it was shown in Tsybakov and Polyak [96] that when problem P has a unique solution x^*, any linear recursive estimation procedure used on problem P satisfies

$$\mathbb{E}\big[(X_k - x^*)(X_k - x^*)^T\big] \geq V k^{-1} + o\big(k^{-1}\big), \tag{19}$$

where $V = B(x^*)^{-1} S (B(x^*)^{-1})^T$, and $B(x^*)$ is the matrix of second derivatives (Hessian) at x^* and where for two $p \times p$ matrices A and B, $A \geq B$ means $A - B$ is positive definite. Hence, any procedure that attains the bound in (19) has been deemed *optimal*. For example, the iterate-averaging procedures Polyak and Juditsky [79] outlined in Section 4.2, and some of their precursors (Fabian [31], Venter [97]), attain the bound in (19) and are hence thought to be optimal. Of course, the bound in (19) is a lower bound and not always attainable, as happens when the underlying function f is nonsmooth.

For optimizing nonsmooth (expectation) functions over the compact convex set $X \subset \mathbb{R}^p$, the bound analogous to (19), given in Nemirovsky and Yudin [69], is stated as follows. Suppose f is a convex and Lipschitz-continuous function satisfying $|f(x) - f(y)| \leq M\|x - y\|, \ \forall x, y \in X$, and denote $f^* := \inf_{x \in X} f(x)$. Then, for $p \geq \mathbb{O}(1)k$, where $\mathbb{O}(1)$ is a universal constant,

$$\mathbb{E}[f(X_k) - f^*] \geq \mathbb{O}(1) M k^{-1/2}. \tag{20}$$

The mirror descent iterate-averaging method (Nemirovskii et al. [68]), outlined in Section 4.2, attains the bound in (20).

Another interpretation of optimality comes from statistics. The famous Cramér–Rao bound (Casella and Berger [19]) implies that a lower bound for any unbiased estimator $T(Y_1, Y_2, \ldots, Y_n)$ of $\theta^* \in \mathbb{R}^p$ constructed using random copies $Y_1, Y_2, \ldots, Y_n$ (not necessarily independent) of a random vector Y having density $g(\cdot; \theta^*)$ satisfies, under certain simple regularity conditions (p. 335 of Casella and Berger [19]) on g, that

$$\mathrm{Var}(T(Y_1, Y_2, \ldots, Y_n)) \geq I(\theta^*)^{-1}; \ I(\theta^*) = -\left(\mathbb{E}\left[\frac{\partial}{\partial \theta^2} \sum_{i=1}^{n} \log g(Y_i; \theta)\right]\right). \tag{21}$$

(The matrix $I(\theta)$ is called the Fisher information matrix at θ.)

To interpret (21) in the context of the current tutorial, suppose the objective function f is strongly convex and twice differentiable. Then any unbiased estimator of x^* that is constructed from k iid observations of the zero gradient at x^* necessarily has variance that exceeds the right-hand side of (21), which in turn can be shown to coincide with Vk^{-1} appearing in (19). It is in this sense that the iterate-averaging procedures (Polyak and Juditsky [79]) outlined in Section 4.2 and some earlier methods (Fabian [31], Venter [97]) placing additional

Table 1. Optimal algorithms for different contexts.

	Nonconvex	Convex	Strongly convex
Smooth	?	RSAG (Ghadimi and Lan [38])	Polyak and Juditsky [79]
Nonsmooth	7	Mirror descent (Nemirovskii et al. [68])	Mirror descent (Nemirovskii et al. [68])

Notes. The algorithm by Polyak and Juditsky [79] and the mirror-descent algorithm are iterate-averaging algorithms, as discussed in Section 4.2. The RSAG (Ghadimi and Lan [38]) algorithm is an accelerated gradient descent algorithm that is not discussed in this tutorial. The optimal bound for the smooth nonconvex context is not known.

stringency are said to be efficient. (Of course, the question of whether there exist biased estimators having mean squared errors lower than the right-hand side of (21) can be answered to be negative except in pathological conditions on the dependence structure of the sample paths.) We are aware of no analogues to (21) in the nonsmooth and nonstrongly convex contexts.

In Table 1 we summarize the optimality bounds in terms of expected optimality gap for the various problem combinations.

Acknowledgments

The authors thank the anonymous referees and Prof. Douglas Shier (Clemson University) for editorial comments on an early draft of the paper. R. Pasupathy was supported in part by the ONR [Grant N000141712295] and National Science Foundation [Grant 1538050].

References

[1] S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Springer, New York, 2007.

[2] A. S. Bandeira, K. Scheinberg, and L. N. Vicente. Convergence of trust-region methods based on probabilistic models. *SIAM Journal on Optimization* 24(3):1238–1264, 2014.

[3] G. Bayraksan and D. P. Morton. A sequential sampling procedure for stochastic programming. *Operations Research* 59(4):898–913, 2009.

[4] A. Beck. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization* 23(4):2037–2060, 2013.

[5] A. Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization* 25(1):185–209, 2015.

[6] A. Beck. *First-Order Methods in Optimization*, MOS-SIAM Series on Optimization. SIAM, Philadelphia, 2017.

[7] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* 2(1):183–202, 2009.

[8] D. P. Bertsekas. Necessary and sufficient conditions for a penalty method to be exact. *Mathematical Programming* 9(1):87–99, 1975.

[9] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[10] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Cambridge, MA, 1996.

[11] D. P. Bertsekas, A. Nedić, and A. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, Belmont, MA, 2003.

[12] D. Blatt, A. O. Hero, and H. Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization* 18(1):29–51, 2007.

[13] R. Bollapragada, R. Byrd, and J. Nocedal. Adaptive sampling strategies for stochastic optimization. Working paper, Northwestern University, Evanston, IL. https://arxiv.org/pdf/1710.11258.pdf, 2017.

[14] V. S. Borkar. Stochastic approximation with two time scales. *Systems and Control Letters* 29(5): 291–294, 1997.

[15] L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review* 60(2):223–311, 2016.

[16] S. Bubeck. Introduction to online optimization. Lecture notes (December 14), Princeton University, Princeton, NJ. https://www.microsoft.com/en-us/research/wp-content/uploads/2017/01/BubeckLectureNotes.pdf, 2011.

[17] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning* 8(3–4):231–358, 2015.

[18] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu. Sample size selection for optimization methods for machine learning. *Mathematical Programming, Series B* 134(1):127–155, 2012.

[19] G. Casella and R. L. Berger. *Statistical Inference*, 2nd ed. Duxbury, Pacific Grove, CA, 2002.

[20] A. Chambolle, R. A. DeVore, N. Y. Lee, and B. J. Lucier. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing* 7(3):319–335, 1998.

[21] C. Charitha, J. Dutta, and D. R. Luke. Lagrange multipliers, (exact) regularization and error bounds for monotone variational inequalities. *Mathematical Programming* 161(1–2):519–549, 2017.

[22] H. Chen and B. W. Schmeiser. Stochastic root finding via retrospective approximation. *IIE Transactions* 33(3):259–275, 2001.

[23] R. Chen, M. Menickelly, and K. Scheinberg. Stochastic optimization using a trust-region method and random models. *Mathematical Programming* 169(2):447–487, 2018.

[24] K. L. Chung. On a stochastic approximation method. *Annals of Mathematical Statistics* 25(3): 463–483, 1954.

[25] P. L. Combettes and J. C. Pesquet. Proximal splitting methods in signal processing. H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, and H. Wolkowicz, eds. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications. Springer, New York, 185–212, 2011.

[26] C. D. Dang and G. Lan. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. *SIAM Journal on Optimization* 25(2):856–881, 2015.

[27] I. Daubechies, M. Defrise, and C. D. Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics* 57(11): 1413–1457, 2004.

[28] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds. *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Red Hook, NY, 1646–1654, 2013.

[29] G. Deng and M. C. Ferris. Variable-number sample-path optimization. *Mathematical Programming* 117(1–2):81–109, 2009.

[30] J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research* 10(December):2899–2934, 2009.

[31] V. Fabian. On asymptotic normality in stochastic approximation. *Annals of Mathematical Statistics* 39(4):1327–1332, 1968.

[32] F. Facchinei and J.-S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems, Vols. I and II*. Springer-Verlag, New York, 2003.

[33] O. Fercoq and P. Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization* 25(4):1997–2023, 2015.

[34] M. C. Ferris and O. L. Mangasarian. Finite perturbation of convex programs. *Applied Mathematics and Optimization* 23(1):263–273, 1991.

[35] M. A. T. Figueiredo and R. D. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing* 12(8):906–916, 2003.

[36] M. P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing* 34(3):A1380–A1405, 2012.

[37] M. P. Friedlander and P. Tseng. Exact regularization of convex programs. *SIAM Journal on Optimization* 18(4):1326–1350, 2008.

[38] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming, Series A* 156(1–2):59–99, 2016.

[39] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, New York, 2004.

[40] J. Hadamard. *Le probleme de Cauchy et les equations aux derivees partielles lineaires hyperbolique*. Hermann, Paris, 1932.

[41] E. Hale, W. Yin, and Y. Zhang. Fixed-point continuation method for ℓ_1-regularized minimization: Methodology and convergence. *SIAM Journal on Optimization* 19(3):1107–1130, 2008.

[42] F. Hashemi, S. Ghosh, and R. Pasupathy. On adaptive sampling rules for stochastic recursions. A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds. *Proceedings of the 2014 Winter Simulation Conference*. IEEE, Piscataway, NJ, 3959–3970, 2014.

[43] F. Hashemi, R. Pasupathy, and M. R. Taaffe. The adaptive sampling gradient method: Optimizing smooth functions with an inexact oracle. Working paper, Virginia Tech, Blacksburg, 2018.

[44] T. Hastie and R. Tibshirani. *Statistical Learning with Sparsity*. Chapman and Hall/CRC, Boca Raton, FL, 2015.

[45] J. Higle and S. Sen. Stochastic decomposition: An algorithm for two-stage stochastic linear programs with recourse. *Mathematics of Operations Research* 16(3):650–669, 1991.

[46] J. Higle and S. Sen. *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1996.

[47] T. Homem-de-Mello. Variable-sample methods for stochastic optimization. *ACM Transactions on Modeling and Computer Simulation* 13(2):108–133, 2003.

[48] M. Hong, X. Wang, M. Razaviyayn, and Z. Q. Luo. Iteration complexity analysis of block coordinate descent methods. *SIAM Journal on Optimization* 163(1–2):85–114, 2017.

[49] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems,* Vol. 26. Curran Associates, Red Hook, NY, 315–323, 2013.

[50] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics* 23(3):462–466, 1952.

[51] S. Kim, R. Pasupathy, and S. G. Henderson. A guide to sample average approximation. M. Fu, ed. *Handbook of Simulation Optimization*, International Series in Operations Research and Management Science. Springer, New York, 2015.

[52] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l_1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing* 1(4):606–617, 2007.

[53] H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, 2003.

[54] T. L. Lai. Stochastic approximation. *Annals of Statistics* 31(2):391–406, 2003.

[55] J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research* 10(December):777–801, 2009.

[56] Y. T. Lee and A. Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science* (*FOCS 2013*). IEEE, Piscataway, NJ, 147–156, 2013.

[57] N. Le Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. Working paper, INRIA, Paris. https://arxiv.org/abs/1607.01375, 2013.

[58] Q. Lin, Z. Lu, and L. Xiao. An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization* 25(4): 2244–2273, 2015.

[59] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis* 16(6):964–979, 1979.

[60] Z. Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications* 72(1):7–35, 1992.

[61] Z. Q. Luo and P. Tseng. Error bounds and convergence analysis of feasible descent methods: A general approach. *Annals of Operations Research* 46(1):157–178, 1993.

[62] N. Majlesinasab, F. Yousefian, and A. Pourhabib. Optimal stochastic mirror descent methods for smooth, nonsmooth, and high-dimensional stochastic optimization. Working paper, Oklahoma State University, Stillwater, https://arxiv.org/abs/1709.08308, 2017.

[63] O. L. Mangasarian and R. R. Meyer. Nonlinear perturbation of linear programs. *SIAM Journal on Control and Optimization* 17(6):745–752, 1979.

[64] J. Mareček, P. Richtárik, and M. Takáč. Distributed block coordinate descent for minimizing partially separable functions. M. Al-Baali, L. Grandinetti, and A. Purnama, eds. *Numerical*

Analysis and Optimization, *Springer Proceedings in Mathematics and Statistics*, Vol. 134. Springer International, Cham, Switzerland, 261–288, 2015.

[65] H. M. Markowitz, G. P. Todd, and W. F. Sharpe. *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. John Wiley & Sons, New York, 2000.

[66] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[67] B. L. Nelson. *Foundations and Methods of Stochastic Simulation: A First Course*. Springer, New York, 2013.

[68] A. Nemirovskii, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization* 19(4):1574–1609, 2009.

[69] A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons, New York, 1983.

[70] Y. E. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/sqr(k))$. *Soviet Mathematics Doklady* 27(2):372–376, 1983.

[71] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science + Business Media, New York, 2004.

[72] Y. E. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization* 22(2):341–362, 2012.

[73] Y. Nesterov and S. Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization* 27(1):110–123, 2017.

[74] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, Berlin, 2006.

[75] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, 1970.

[76] R. Pasupathy. On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization. *Operations Research* 58(4, Part 1):889–901, 2010.

[77] R. Pasupathy and B. W. Schmeiser. Retrospective-approximation algorithms for multidimensional stochastic root-finding problems. *ACM Trans. Modeling Comput. Simulation* 19(2):1–36, 2009.

[78] B. Polyak. *Introduction to Optimization*. Optimization Software, New York, 1987.

[79] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization* 30(4):838–855, 1992.

[80] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review* 52(3):471–501, 2012.

[81] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming* 144(2):1–38, 2014.

[82] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics* 22(3):400–407, 1951.

[83] L. Rosasco, S. Villa, and B. C. Vu. Stochastic forward-backward splitting for monotone inclusions. *Journal of Optimization Theory and Applications* 169(2):388–406, 2016.

[84] J. Sacks. Asymptotic distribution of stochastic approximation procedures. *Annals of Mathematical Statistics* 29(2):373–405, 1958.

[85] M. Schmidt, N. Le Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, eds. *Proceedings of the 24th International Conference on Neural Information Processing Systems*. Curran Associates, Red Hook, NY, 1458–1466, 2011.

[86] S. Shalev-Shwartz and A. Tewari. Stochastic methods for ℓ_1-regularized loss minimization. *Journal of Machine Learning Research* 12(February):1865–1892, 2011.

[87] S. Shalev-Shwartz and A. Tewari. Online learning and online convex optimization. *Foundations and Trends in Machine Learning* 4(2):107–194, 2012.

[88] A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, Philadelphia, 2009.

[89] S. Shashaani, F. S. Hashemi, and R. Pasupathy. ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free simulation optimization. Working paper, Purdue University, Lafayette, IN, 2016.

[90] S. Shashaani, S. R. Hunter, and R. Pasupathy. ASTRO-DF: Adaptive sampling trust-region optimization algorithms, heuristics, and numerical experience. T. M. K. Roeder, P. I. Frazier,

R. Szechtman, and E. Zhou, eds. *Proceedings of the 2016 Winter Simulation Conference*. IEEE, Piscataway, NJ, 554–565, 2016.

[91] R. Shefi and M. Teboulle. Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization. *SIAM Journal on Optimization* 24(1): 269–297, 2014.

[92] N. Srebro, J. D. M. Rennie, and T. S. Jaakola. Maximum-margin matrix factorization. L. K. Saul, Y. Weiss, and L. Bottou, eds. *Advances in Neural Information Processing Systems*, Vol. 17. MIT Press, Cambridge, MA1329–1336, 2005.

[93] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society* 58(1):267–288, 1996.

[94] A. N. Tikhonov, A. V. Goncharsky, V. V. Stepanov, and A. G. Yagola. *Numerical Methods for the Solution of Ill-Posed Problems*, *Mathematics and Its Applications*. Springer-Science+Business Media, Dordrecht, Netherlands, 1995.

[95] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming, Series B* 117(1):387–423, 2009.

[96] A. Tsybakov and B. T. Polyak. Optimal order of accuracy of search algorithms in stochastic optimization. *Problemy Peredachi Informatsii* 26(2):45–53, 1990.

[97] H. J. Venter. An extension of the Robbins-Monro procedure. *Annals of Mathematical Statistics* 38(1):181–190, 1967.

[98] S. J. Wright. Accelerated block-coordinate relaxation for regularized optimization. *SIAM Journal on Optimization* 22(1):159–186, 2012.

[99] S. J. Wright. Coordinate descent algorithms. *Mathematical Programming, Series B* 151(1):3–34, 2015.

[100] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP 2008*). IEEE, Piscataway, NJ, 3373–3376, 2008.

[101] Y. Xu and W. Yin. A block coordinate descent method for multi-convex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences* 6(3):1758–1789, 2013.

[102] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics* 4(5):1–17, 1964.

[103] T. Yang, Q. Lin, and Z. Li. Unified convergence analysis of stochastic momentum methods for convex and non-convex optimization. Working paper, University of Iowa, Iowa City, http://arxiv.org/abs/1604.03257, 2016.

[104] F. Yousefian, A. Nedić, and U. V. Shanbhag. On stochastic gradient and subgradient methods with adaptive steplength sequences. *Automatica* 48(1):56–67, 2012.

[105] F. Yousefian, A. Nedić, and U. V. Shanbhag. On stochastic mirror-prox algorithms for stochastic Cartesian variational inequalities: Randomized block coordinate and optimal averaging schemes. *Set-Valued and Variational Analysis*, ePub ahead of print March 20, https://doi.org/10.1007/s11228-018-0472-9, 2018.