# Chapter 5

# Stochastic Gradient

The stochastic gradient (SG) method is one of the most popular algorithms in modern data analysis and machine learning. It has a long history, with variants having been invented and reinvented several times by different communities, under such names as "least mean squares," "back propagation," "online learning," and the "randomized Kaczmarz method." Most people attribute the stochastic gradient approach to the 1951 work of Robbins and Monro [84], who were interested in devising efficient algorithms for computing random means and roots of scalar functions for which only noisy values are available. In this chapter, we explore some of the properties and implementation details of SG.

As in much of this book, our goal is to minimize the multivariate convex function $f : \mathbb{R}^n \to \mathbb{R}$, which we assume here to be smooth for purposes of this discussion. Extension of SG to the case of *nonsmooth* convex functions is straightforward, and left as an exercise in the chapter on nonsmooth methods. SG differs from methods of Chapters 3 and 4 in the kind of information that is available about $f$. In place of an exact value of $\nabla f(x)$, we assume that we can compute or acquire a vector $g(x, \xi) \in \mathbb{R}^n$, which is a function of a random variable $\xi$ as well as $x$, such that

$$\nabla f(x) = \mathbb{E}_\xi[g(x, \xi)]. \tag{5.1}$$

We assume that $\xi$ belongs to some space $\Xi$ with distribution $P$, and $\mathbb{E}_\xi$ denotes the expectation taken over $\xi \in \Xi$ according to distribution $P$. Equation (5.1) asserts that $g(x, \xi)$ is an *unbiased* estimate of $\nabla f(x)$. SG proceeds by substituting $g(x, \xi)$ for the true gradient $\nabla f$ in the steepest-descent update formula, so each iteration is as follows:

$$x^{k+1} = x^k - \alpha_k g(x^k, \xi^k), \tag{5.2}$$

where the random variable $\xi^k$ is chosen according to the distribution $P$ (independently of the choices at other iterations) and $\alpha_k > 0$ is the steplength. The method steps in a direction that *in expectation* equals the steepest descent direction. Although $g(x^k, \xi^k)$ may differ substantially from $\nabla f(x^k)$ — it may contain a lot of "noise" — it also contains enough "signal" to make progress toward the optimal value of $f$ over the long term. In typical applications, computation of the gradient estimate $g(x^k, \xi^k)$ is much cheaper than computation of the true gradient $\nabla f(x^k)$.

The choice of steplength $\alpha_k$ is critical to the theoretical and practical behavior of SG. We cannot expect to match the performance of steepest descent, in which we move along the true negative gradient direction $-\nabla f(x^k)$ rather than its noisy approximation $-g(x^k, \xi^k)$. In the steepest descent

method, the constant steplength $\alpha_k \equiv 1/L$ (where $L$ is the Lipschitz constant for $\nabla f$) yields convergence; see Chapter 3. We can show that this constant-steplength choice will not yield the same convergence properties in the stochastic gradient context, by considering what happens if we initialize the method at the minimizer of $f$, that is, $x^0 = x^*$. Since $\nabla f(x^*) = 0$, there are no descent directions, and the methods of Chapter 3 will generate a zero step — as they should, since we are already at a solution. The stochastic gradient direction $g(x^0, \xi^0)$ may however be nonzero, causing SG to step away from the solution (and increase the objective). We can show however that for judicious choice of the steplength sequence $\{\alpha_k\}$, the sequence $\{x^k\}$ converges to $x^*$, or at least to a neighborhood of $x^*$, at rates that are typically slower than those achieved by (true-)gradient descent.

## 5.1 Examples and Motivation

There are many situations in which SG is a powerful tool. Here we discuss a few motivating examples that drive our subsequent implementation details and theoretical analyses.

### 5.1.1 Noisy Gradients

The simplest application of SG is to the case when the gradient estimate $g(x, \xi)$ is the true gradient with additive noise, that is,

$$g(x, \xi) = \nabla f(x) + \xi, \tag{5.3}$$

where $\xi$ is some noise process. The unbiasedness property (5.1) will hold provided that $\mathbb{E}(\xi) = 0$. Our analysis below reveals a protocol for choosing step sizes $\alpha_k$ so that SG (5.2) converges. Formula (5.2) reduces in this case to

$$x^{k+1} = x^k - \alpha_k(\nabla f(x^k) + \xi^k), \tag{5.4}$$

which is steepest descent with an additive noise term $\alpha_k \xi^k$.

### 5.1.2 Incremental Gradient Method

The incremental gradient method, also known as the perceptron or back-propagation, is one of the most common variants of SG. Here we assume that $f$ has the form of a finite sum, that is,

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x). \tag{5.5}$$

where $n$ is usually very large. Computing a full gradient $\nabla f$ generally requires computation of $\nabla f_i$, $i = 1, 2, \ldots, N$ — a computation that scales proportionally to $n$ in general. Iteration $k$ of the incremental gradient procedure selects some index $i_k$ from $\{1, 2, \ldots, N\}$ and sets

$$x^{k+1} = x^k - \alpha_k \nabla f_{i_k}(x^k).$$

That is, we choose one of the functions $f_i$ and follow its negative gradient. The standard incremental gradient method chooses $i_k$ to cycle through the components $\{1, 2, \ldots, N\}$ in order, that is, we set $i_k = (k \mod N) + 1$ for $k = 0, 1, 2, \ldots$.

Alternatively, we could choose $i_k$ according to some random procedure at each iteration, which would be an SG approach. We see this by defining the random variable space $\Xi$ to be the set of indices $\{1, 2, \ldots, N\}$, and the choice of random variable $\xi^k$ is the index $i_k \in \{1, 2, \ldots, N\}$, so that $g(x^k, \xi^k) = \nabla f_{i_k}(x^k)$. Here, the distribution $P$ is such that $P(i) = 1/N$ for all $i = 1, 2, \ldots, N$. The unbiasedness property (5.1) holds, since

$$E_\xi(g(x, \xi)) = \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(x) = \nabla f(x).$$

The convergence analysis of the SG method just described are straightforward, as we will see. Surprisingly, analysis of standard incremental gradient, with the cyclic choice of indices $i_k$, is more challenging, and the convergence guarantees are weaker.

### 5.1.3 Classification and the Perceptron

Classification is a canonical problem in machine learning, as we showed in Chapter 1. We have data consisting of pairs $(a_i, y_i)$, with feature vectors $a_i \in \mathbb{R}^n$ and labels $y_i \in \{-1, 1\}$ for $i = 1, 2, \ldots, N$. The goal is to find a vector $x \in \mathbb{R}^n$ such that

$$x^T a_i > 0 \text{ for } y_i = 1, \quad x^T a_i < 0 \text{ for } y_i = -1.$$

Any $x$ satisfying these requirements defines a line through the origin with all positive examples on one side and all negative examples on the other. (Often, the division is not so clean, in that the data may allow no line to perfectly separates the two classes, but we can still search for a $w$ that most nearly achieves this goal.)

A popular algorithm for finding $x$ called the *perceptron* was invented in the 1950s. It uses one example at a time to generate a sequence $\{x^k\}$, $k = 1, 2, \ldots$ from some starting point $x^0$. At iteration $k$, we choose one of our data pairs $(a_{i_k}, y_{i_k})$ and update according to the formula

$$x^{k+1} = (1 - \gamma)\, x^k + \begin{cases} \eta y_{i_k} a_{i_k} & \text{if } y_{i_k} (x^k)^T a_{i_k} < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{5.6}$$

for some positive parameters $\gamma$ and $\eta$. If the current guess $x^k$ classifies the pair $(a_{i_k}, y_{i_k})$ incorrectly, then this iteration "nudges" $x^k$ to make $(x^k)^T a_{i_k}$ closer to the correct sign. If $x^k$ produces correct classification on this example, no change is made.

This method is an instance of SG. A quick calculation shows that this procedure is obtained by applying SG to the cost function

$$\frac{1}{N} \sum_{i=1}^{N} \max\left(1 - y_i a_i^T x, 0\right) + \frac{\lambda}{2} \|x\|_2^2, \tag{5.7}$$

where $\xi^k$ is the index $i_k$ of a single term from the summation. In the update equation (5.6), we have used (5.2) with

$$g(x^k, \xi^k) = g(x^k, i_k) = \lambda x^k + \begin{cases} -\eta y_{i_k} a_{i_k} & \text{if } y_{i_k} (x^k)^T a_{i_k} < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{5.8}$$

and $\gamma = \alpha_k \lambda$ and $\eta = \alpha_k$. (In machine learning, the stepsize is often referred to as the *learning rate*.) The cost function (5.7) is often called the *support vector machine* (see Section 1.4). In the parlance of our times, the perceptron algorithm is equivalent to "training" a support vector machine using SG.

### 5.1.4 Empirical Risk Minimization

In machine learning, the support vector machine is one of many instances of the class of optimization problems called *Empirical Risk Minimization*. Many classification, regression, and decision tasks can be evaluated as expected values of error over the distribution form which the data is drawn or generated. The most common example is known as *statistical risk*. Given a data generating distribution $P$, and a *loss function* $\ell(u, v)$ we define the risk as

$$R[f] := \mathbb{E}_{(x,y)\sim P}\, \ell(f(x), y), \tag{5.9}$$

that is, the expectation is taken over the data space $(x, y)$ according to probability distribution $P$. The function $\ell$ measures the cost of assigning the value $f(x)$ when the quantity to be estimated is $y$. (Typically, $\ell$ becomes larger when $f(x)$ deviates further from $y$.) The quantity $R$ is the expected loss of the decision rule $f(x)$ with respect to the probability distribution $P$. The goal of many learning tasks is to choose the function $f$ that minimizes the risk. For example, the support vector machine uses a "hinge loss" as the function $\ell$, that measures the distance between the prediction $w^T x$ and the correct half-space. In regression problems, $y$ is a target variate, and the loss measures the difference between $f(x)$ and $y$ according to the square function $\ell(f(x), y) = \frac{1}{2}(f(x) - y)^2$.

Often, minimization (and even evaluation) of the risk function (5.9) is computationally intractable and requires knowledge of the likelihood and prior models for the data pairs $(x, y)$. A popular alternative uses *samples* to provided an estimate for the true risk. Suppose we have a process that generates independent, identically distributed (i.i.d.) samples $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ from the joint distribution $p(x, y)$. For these data points and a fixed decision rule $\hat{x}(y)$, we can expect that *the empirical risk* defined by

$$R_{\mathrm{emp}}[f] := \frac{1}{N} \sum_{i=1}^{N} \ell(f(y_i), x_i) \tag{5.10}$$

to be "close" to the true risk $R[f]$. Indeed, $R_{\mathrm{emp}}[f]$ is a random variable equal to the sample mean of the loss function. If we take the expectation with respect to our sample set, we have

$$\mathbb{E}\, R_{\mathrm{emp}}[f] = R[f]\,.$$

Given these samples, the empirical risk is no longer a function of the likelihood and prior models. It yields a simpler optimization problem, in which the objective is a finite sum of the form (5.5). Minimizing this empirical risk corresponds to finding the best function $f$ that minimizes the average loss over our data sample.

SG and empirical risk minimization (ERM) are intimately related. One variant of ERM formulates the problem finitely as (5.10) and then applies the randomized incremental gradient approach of Section 5.1.2 to this function. Another variant does not explicitly take a finite data sample, instead applying SG directly to (5.9). At each step, a pair $(x, y)$ is sampled according to the distribution $p(x, y)$, and a step is taken along the negative gradient of loss function $\ell$ with respect to $f$, evaluated at the point $(f(x), y)$.

The perceptron algorithm is a particular instance of ERM, in which we define $f(x) = w^T x$ (so that $f$ is parametrized by the vector $w$) and $\ell(f(x), y) = \max(1 - yx^T w, 0)$.

## 5.2 Randomness and Steplength: Insights

Before turning to a rigorous analysis of SG, we give some background and insight into how to choose the stepsize parameters $\alpha_k$, using some simple but informative examples.

### 5.2.1 Example: Computing a Mean

Consider applying an incremental gradient method to the scalar function

$$f(x) := \frac{1}{2N} \sum_{i=1}^{N} (x - \omega_i)^2 \tag{5.11}$$

where $\omega_i$, $i = 1, 2, \ldots, N$ are fixed scalars. This function has the form of the finite sum (5.5) when we define $f_i(x) = \frac{1}{2}(x - \omega_i)^2$, so that

$$\nabla f_i(x) = x - \omega_i \,.$$

We start with $x^0 = 0$ and, as in standard incremental gradient, step through the indices in order and use the stepsize $\alpha_k = 1/(k+1)$. The first few iterations are:

$$\begin{aligned} x^1 &= x^0 - (x^0 - \omega_1) = \omega_1, \\ x^2 &= x^1 - \frac{1}{2}\left(x^1 - \omega_2\right) = \frac{1}{2}\omega_1 + \frac{1}{2}\omega_2, \\ x^3 &= x^2 - \frac{1}{3}\left(x^2 - \omega_3\right) = \frac{1}{3}\omega_1 + \frac{1}{3}\omega_2 + \frac{1}{3}\omega_3, \end{aligned}$$

so that

$$x^k = \left(\frac{k-1}{k}\right) x^{k-1} + \frac{1}{k}\omega_k = \frac{1}{k} \sum_{j=1}^{k} \omega_j \,, \quad k = 1, 2, \ldots \,. \tag{5.12}$$

The stepsize $\alpha_k = 1/(k+1)$ was the one originally proposed by Robbins and Monro [84], and it makes perfect sense for this simple example, as it produces iterates that are the running average of all the samples $\omega_j$ encountered so far. This choice of step size has two other important features.

- Even when the gradients $g(x; i) = \nabla f_i(x)$ are bounded in norm, the iterates can traverse an arbitrary distances across the search space, because $\sum_{k=0}^{\infty} 1/(k+1) = \infty$. Thus, convergence can be obtained even when the starting point $x^0$ is arbitrarily far from the solution $x^*$.
- The steplengths shrink to zero, so that when the iterates reach a neighborhood of the solution $x^*$, they tend to stay there, even though the search directions $g(x; \xi)$ contain noise.

For this simple example, the global minimum of $f$ is found after $N$ steps of the cyclic, incremental method; there is no need for randomness. In fact, when we choose the component function $f_{i_k}$ randomly, we are unlikely to converge to the minimizer of (5.11) in finite number of iterations.

However, there are other instances of finite-sum objectives in which randomness produces much better performance than cyclic schemes, as we see in the next section.

Let us consider now a "continuous" version of (5.11):

$$f(x) = \tfrac{1}{2}\mathbb{E}_{\omega}\,(x-\omega)^2, \tag{5.13}$$

where $\omega$ is some random variable with mean $\mu$ and variance $\sigma^2$. At step $j$ of SG, we select some value $\omega_{j+1}$ from the distribution of $\omega$, independently of the choices of $\omega$ that were made at previous iterations. We take a step of length $1/(j+1)$ in direction $x^j - \omega_{j+1}$. After $k$ steps, starting from $x^0 = 0$, we have as before that $x^k$ satisfies (5.12). By plugging this value into (5.13), and taking the expectation over $\omega$ and all the random variables $\omega_1, \omega_2, \ldots, \omega_k$, we obtain

$$f(x^k) = \frac{1}{2}\mathbb{E}_{\omega_1,\omega_2,\ldots,\omega_k,\omega}\left[\left(\frac{1}{k}\sum_{j=1}^{k}\omega_j - \omega\right)^2\right] = \frac{1}{2k}\sigma^2 + \frac{1}{2}\sigma^2. \tag{5.14}$$

In this simple case too, we can compute the minimizer of (5.13) exactly. We have

$$f(x) = \tfrac{1}{2}\mathbb{E}\left[x^2 - 2\omega x + \omega^2\right] = \tfrac{1}{2}x^2 - \mu x + \tfrac{1}{2}\sigma^2 + \tfrac{1}{2}\mu^2\,.$$

Thus the minimizer of $f$ is $x^* = \mu$, with $f(x^*) = \frac{1}{2}\sigma^2$. By comparing this value with (5.14), we have

$$f(x^k) - f(x^*) = \frac{1}{2k}\sigma^2.$$

Statistically speaking, it can be shown that $x^k$ is the highest-quality estimate that can be attained for $x^*$ given the sequence $\{\omega_1, \omega_2, \ldots, \omega_k\}$. Interestingly, SG, which considered the samples $\omega_{j+1}$ one at a time and made a step after each iteration, is able to achieve the same quality as an estimator that made use of the complete set of data $\{\omega_1, \omega_2, \ldots, \omega_k\}$ at once. Even so, the convergence rate for this best-possible performance is sublinear: The sequence of differences between function values and their optimum $\{f(x^k) - f^*\}$ shrinks like $1/k$, rather than decreasing exponentially to zero. This demonstrates a fundamental limitation of SG: Linear convergence cannot be expected in general. *Statistics*, not computation or algorithm design, stands in the way of linear convergence rates.

### 5.2.2 The Randomized Kaczmarz Method

The potential benefits of randomness can be seen when we consider a special case of the following linear least squares problem:

$$\min\, f(x) := \frac{1}{2N}\sum_{i=1}^{N}\left(a_i^T x - b_i\right)^2, \tag{5.15}$$

where $\|a_i\| = 1$, $i = 1, 2, \ldots, N$. Assume that there exists an $x^*$ such that $a_i^T x^* = b_i$ for $i = 1, 2, \ldots, N$. This point will be a minimizer of $f$, with $f(x^*) = 0$. SG with stepsize $\alpha_k \equiv 1$ — known as the *randomized Kaczmarz* method — yields the recursion

$$x^{k+1} = x^k - a_{i_k}\left(a_{i_k}^T x^k - b_{i_k}\right) = x^k - a_{i_k}a_{i_k}^T(x^k - x^*).$$

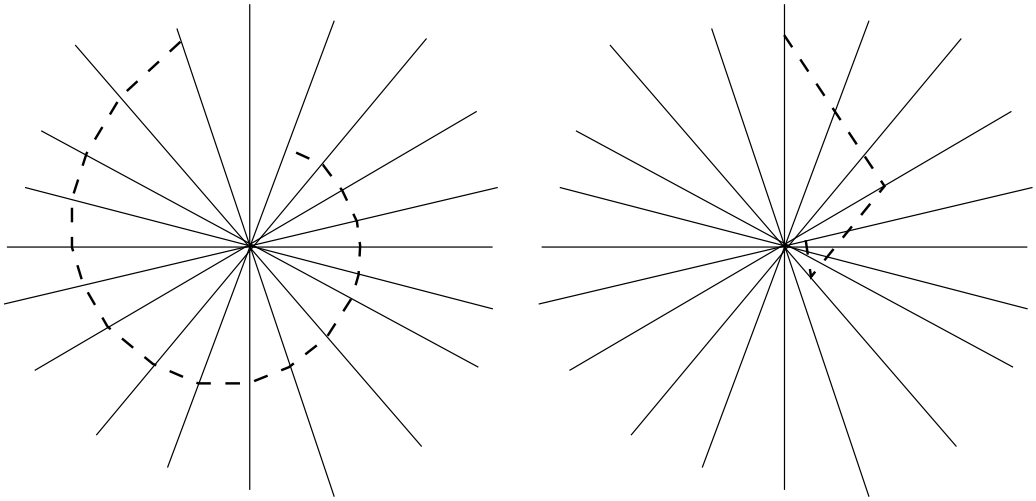

Figure 5.1: Kaczmarz method. Deterministic, ordered choice (left) leads to slow convergence; randomized Kaczmarz (right) converges faster.

Aggregating the effects of the first $k$ iterations, we obtain

$$x^{k+1} - x^* = \left(I - a_{i_k} a_{i_k}^T\right)\left(x^k - x^*\right) = \prod_{j=0}^{k}\left(I - a_{i_j} a_{i_j}^T\right)\left(x^0 - x^*\right).$$

Iteration $k$ is a projection of the current iterate $x^k$ onto the plane defined by $a_{i_k}^T x = b_{i_k}$. If two successive subspaces are close to one another, then $x^{k+1}$ and $x^k$ are close together, and we do not make much progress toward $x^*$. The following example describes a set of vectors $\{a_1, a_2, \ldots, a_N\}$ such that the deterministic, cyclic choice of indices $i_k = (k \mod N) + 1$ yields slow progress, while much faster convergence is attained by making random choices of $i_k \in (1, 2, \ldots, N\}$ for each $k$.

For $N \geq 3$, set $\omega_N := \pi/N$ and define the vectors $a_i$ as follows:

$$a_i = \begin{bmatrix} \cos(i\omega_N) \\ \sin(i\omega_N) \end{bmatrix}, \quad i = 1, 2, \ldots, N. \tag{5.16}$$

Define $b_i = 0$, $i = 1, 2, \ldots, N$, so that the solution of (5.15) is $x^* = 0$. We have that $\|a_i\| = 1$ for all $i$, and in addition that $\langle a_i, a_{i+1}\rangle = \cos(\omega_N)$ for $1 \leq i \leq N-1$. The matrices $M_i := I - a_i a_i^T$ are positive semidefinite for all $i$, and the following identity is satisfied:

$$\mathbb{E}_j(M_j) = \frac{1}{N}\sum_{i=1}^{N} M_i = \frac{1}{2} I. \tag{5.17}$$

Any set of unit vectors satisfying (5.17) is called a *normalized tight frame*, and the vectors (5.16) form a *harmonic frame*, due to their trigonometric origin.

Consider a *randomized* version of the Kaczmarz method, in which we select the vector $a_{i_k}$ with equal likelihood from among $\{a_1, a_2, \ldots, a_N\}$, with the choice made independently at each iteration.

The expected decrease in error over iteration $k$, conditional on the value of $x^k$, is

$$\mathbb{E}_{i_k}(x^{k+1} - x^* \,|\, x^k) = \left(\mathbb{E}_{i_k}(I - a_{i_k} a_{i_k}^T)\right)(x^k - x^*) = \tfrac{1}{2}(x^k - x^*), \tag{5.18}$$

where we used (5.17) to obtain the fraction of 1/2. The following argument shows exponential decrease of the expected error with rate (1/2) per iteration:

$$\begin{aligned}
&\mathbb{E}(x^k - x^0) \\
&= \mathbb{E}_{i_0,i_1,\ldots,i_{k-1}} \prod_{j=0}^{k-1} M_{i_j}(x^0 - x^*) = \left[\prod_{j=0}^{k-1} \mathbb{E}_{i_j}(M_{i_j})\right](x^0 - x^*) = \left[\mathbb{E}_{i_j}(M_{i_j})\right]^k (x^0 - x^*) = 2^{-k}(x^0 - x^*)\,.
\end{aligned}$$

(The critical step of taking the expectation inside the product is possible because of independence of the $i_j$, $j = 0, 1, \ldots, k-1$.)

The behavior of randomized Kaczmarz is shown in the right diagram in Figure 5.1, with the path traced by the iterations shown as a dotted line.

Why do we attain linear convergence for the randomized method, when the example of computing means in Section 5.2.1 attained only a sublinear rate? The answer is that this problem is rather special, in that the solution $x^*$ is a fixed point of both a gradient map *and* a stochastic gradient step. That is, both $\nabla f(x)$ and $\nabla f_i(x)$ approach zero as $x \to x^*$, for all $i = 1, 2, \ldots, N$. For the same reason, we were able to use a large constant stepsize $\alpha_k \equiv 1$ rather than the usual decreasing stepsize.

The fact that the vectors $a_{i_k}$ are selected randomly, for $k = 0, 1, 2, \ldots$ is also critical to the fast convergence. If we use the deterministic order $i_k = k + 1$, $k = 0, 1, 2, \ldots, N-1$, the convergence analysis is quite different. Define the vectors

$$\hat{a}_i = \begin{bmatrix} \sin(-i\omega_N) \\ \cos(-i\omega_N) \end{bmatrix},$$

and note that $M_i = I - a_i a_i^T = \hat{a}_i \hat{a}_i^T$. Since $\langle \hat{a}_i, \hat{a}_{i+1} \rangle = \cos(\omega_N)$, it follows that

$$\prod_{i=1}^{k} M_i = \hat{a}_k \hat{a}_1^T \prod_{j=1}^{k-1} \langle \hat{a}_j, \hat{a}_{j+1} \rangle = \hat{a}_k \hat{a}_1^T \cos^{k-1}(\omega_N)\,.$$

We therefore have

$$\|x^k - x^*\| = \left\| \prod_{i=1}^{k} \left(I - a_i a_i^T\right)\left(x^0 - x^*\right) \right\| = \cos^{k-1}(\omega_N) \left| \hat{a}_1^T \left(x^0 - x^*\right) \right|\,.$$

For $x^0 = (0, 1)^T$, we have $\hat{a}_1^T (x^0 - x^*) = \|x^0 - x^*\|$, so

$$\|x^k - x^*\| = \cos(-\pi/N)^{k-1} \|x^0 - x^*\|, \quad k = 0, 1, 2, \ldots, N.$$

This indicates geometric convergence, at a rate of $\cos(-\pi/N) \approx 1 - (1/2)(\pi/N)^2$ per iteration — a much slower rate than the linear rate of 1/2 achieved in the randomized case. (Note however that the analysis for the cyclic case is deterministic, whereas the faster convergence rate in the stochastic case is for the *excepted* error.)

The deterministic variant is plotted in the left diagram of Figure 5.1, showing a slow spiral toward the solution.

The randomized Kaczmarz method was analyzed some years ago by signal processing researchers, independently of SG developments.

## 5.3 Convergence Analysis: Key Assumptions

We now turn to convergence analysis of SG, applied to the convex function $f : \mathbb{R}^n \to \mathbb{R}$, with steps of the form (5.2) and search directions $g(x, \xi)$ satisfying condition (5.1). To prove convergence, we need to assume some bounds on the sizes of the gradient estimates $g(x, \xi)$, so that the information they contain is not swamped by noise. We assume that there are nonnegative constants $L_g$ and $B$ such that

$$\mathbb{E}_\xi, \left[\|g(x;\xi)\|_2^2\right] \leq L_g^2 \|x - x^*\|^2 + B^2 \quad \text{for all } x. \tag{5.19}$$

Note that this assumption may be satisfied even when $g(x;\xi)$ is arbitrarily large for some combination of $x$ and $\xi$; formula (5.19) requires only boundedness *in expectation over* $\xi$ for each $x$. (Section 5.3.3 below contains an example in which $\xi$ is unbounded but (5.19) still holds for suitable choices of $L_g$ and $B$.)

Note that when $L_g = 0$ in (5.19), $f$ cannot be strongly convex over an unbounded domain. If $f$ were strongly convex function with modulus of convexity $m$, we would have

$$\|\nabla f(x)\| \geq \frac{m}{2}\|x - x^*\|$$

for all $x$. On the other hand, we have by Jensen's inequality that

$$\|\nabla f(x)\|^2 = \|\mathbb{E}\, g(x;\xi)\|^2 \leq \mathbb{E}\,[\|g(x;\xi)\|^2].$$

These two bounds together imply that it is not possible to find a $B$ for which (5.19) holds with $L_g = 0$, if the domain of $f$ is unbounded.

When $f$ has the finite-sum form (5.5) and we have $\nabla f_{i_k}(x^k)$ as the gradient estimate at iterate $x^k$, where $i_k$ chosen uniformly at random from $\{1, 2, \ldots, N\}$, as in Section 5.1.2, the bound (5.19) specializes to

$$\frac{1}{N}\sum_{i=1}^{N} \|\nabla f_i(x)\|^2 \leq L_g^2 \|x - x^*\|^2 + B^2 \quad \text{for all } x. \tag{5.20}$$

The steplengths $\alpha_k$ in the stochastic gradient iteration formula (5.2) typically depend on the constants $L_g$ and $B$ in (5.19). Throughout, we will assume that the sequence $\{\xi^k\}_{k=0,1,2,\ldots}$ needed to generate the gradient approximations $g(x^k, \xi^k)$ is selected i.i.d. from a fixed distribution. (It is possible to weaken the i.i.d. assumptions, but we do not consider such extensions here.)

We now examine how the constants $L_g$ and $B$ appear in different problem settings, including those described in earlier sections.

### 5.3.1 Case 1: Bounded Gradients: $L_g = 0$

Suppose that the stochastic gradient function $g(\cdot;\cdot)$ is bounded almost surely for all $x$ — that is, $L_g = 0$ in (5.19). This is true for the *logistic regression* objective

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} -y_i x^T a_i + \log(1 + \exp(x^T a_i)), \tag{5.21}$$

where the data are $(a_i, y_i)$ with $y_i \in \{0, 1\}$, $i = 1, 2, \ldots, N$. Following the finite-sum setting (5.5), the random variable $\xi$ is drawn uniformly from the set $\{1, 2, \ldots, N\}$, and

$$g(x; i) = \left(-y_i + \frac{\exp(x^T a_i)}{1 + \exp(x^T a_i)}\right) a_i$$

Thus (5.19) holds with $L_g = 0$ and $B = \sup_{i=1,2,\ldots,N} \|a_i\|_2$.

### 5.3.2 Case 2: Randomized Kaczmarz: $B = 0$, $L_g > 0$

Consider the least-squares objective (5.15), where we assume that $a_i \neq 0$ but not necessarily $\|a_i\| = 1$ for each $i$. Assume that there is $x^*$ for which $f(x^*) = 0$, that is, $a_i^T x^* = b_i$ for all $i = 1, 2, \ldots, N$. By substituting into (5.15), we obtain

$$f(x) = \frac{1}{2N} \sum_{i=1}^{N} (x - x^*)^T a_i a_i^T (x - x^*)$$

and, with the random variable $\xi$ being drawn uniformly from $\{1, 2, \ldots, N\}$, we have

$$g(x; i) = a_i a_i^T (x - x^*).$$

For the expected norm, we have

$$\mathbb{E}\left[\|g(x;i)\|^2\right] = \mathbb{E}\left[\|a_i\|^2 |a_i^T(x - x^*)|^2\right] \leq \mathbb{E}\left[\|a_i\|^4\right] \|x - x^*\|^2,$$

so that (5.19) can be satisfied by setting $L_g = \mathbb{E}[\|a_i\|^4]^{1/2}$ and $B = 0$.

### 5.3.3 Case 3: Additive Gaussian Noise

Consider the additive noise model (5.3) where $\xi$ is from the Gaussian distribution with mean zero and covariance $\sigma^2 I$, that is, $\xi \in N(0, \sigma^2 I)$. We have $\mathbb{E}\left[g(x;\xi)\right] = \nabla f(x)$ and

$$\mathbb{E}[\|g(x;\xi)\|^2] = \|\nabla f(x)\|^2 + 2\nabla f(x)^T \mathbb{E}(\xi) + \mathbb{E}(\|\xi\|^2) = \|\nabla f(x)\|^2 + n\sigma^2. \tag{5.22}$$

We can satisfy (5.19) by setting $B = \sigma\sqrt{n}$, and defining $L_g$ to the Lipschitz constant $L$ of the gradient of $f$ (because $\|\nabla f(x)\|^2 \leq \|\nabla f(x) - \nabla f(x^*)\|^2 \leq L^2 \|x - x^*\|^2$).

### 5.3.4 Case 4: Incremental Gradient

Consider the finite-sum formulation (5.5) in which the gradient $\nabla f_i$ of each term in the sum has Lipschitz constant $L_i$. As in Section 5.1.2, the distribution for the random variable $\xi$ is discrete with $N$ equally likely choices corresponding to the indices $i = 1, 2, \ldots, N$ of the terms in the sum. For the $i$th term $f_i(x)$, we define $x^{*i}$ to be any point for which $\nabla f_i(x^{*i}) = 0$. We then have

$$\begin{aligned}
\mathbb{E}_\xi[\|g(x;\xi)\|^2] &= \mathbb{E}_i[\|\nabla f_i(x)\|^2] \\
&\leq \mathbb{E}[L_i^2 \|x - x^{*i}\|^2] \\
&\leq \mathbb{E}\left[2L_i^2 \|x - x^*\|^2 + 2L_i^2 \|x^{*i} - x^*\|^2\right] \\
&= \frac{2}{N} \sum_{i=1}^{N} L_i^2 \|x - x^*\|^2 + \frac{2}{N} \sum_{i=1}^{N} L_i^2 \|x^{*i} - x^*\|^2,
\end{aligned}$$

where we used the bound $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$. Thus (5.19) holds if we define

$$L_g^2 = \frac{2}{N} \sum_{i=1}^{N} L_i^2, \quad B^2 = \frac{2}{N} \sum_{i=1}^{N} L_i^2 \|x^{*i} - x^*\|^2.$$

There is nice intuition for this choice of $B$. If $x^{*i} = x^*$ for all $i$, then $B = 0$, as in the case of Randomized Kaczmarz (Section 5.3.2).

## 5.4 Convergence Analysis

Our convergence results track the decrease in certain measures of error as a function of iteration count. These measures are of two types. The first is an expected squared error in the point $x$, that is $\mathbb{E}\left[\|x - x^*\|^2\right]$, where $x^*$ is the solution and the expectation is taken over all the random variables $\xi^k$ encountered to that point of the algorithm. This measure is most appropriate when the objective $f$ is strongly convex, so that the solution $x^*$ is uniquely defined. The second measure of optimality is the gap between the current objective value and the optimal value, that is $f(x) - f(x^*)$. This measure can be used when $f$ is convex but not necessarily strongly convex. In the strongly convex case, each of these two measures can be bounded in terms of the other, with the bound depending on the Lipschitz constant for $\nabla f$ and the modulus of convexity $m$.

We see that the suitable choices of steplengths $\alpha_k$ in (5.2) depend on $L_g$ and $B$, and that the convergence rates also depend on these two quantities.

Using the update formula (5.2), we expand the distance to the optimal solution, as follows:

$$\begin{aligned}\|x^{k+1} - x^*\|^2 &= \|x^k - \alpha_k g(x^k;\xi^k) - x^*\|^2 \\ &= \|x^k - x^*\|^2 - 2\alpha_k \langle g(x^k;\xi^k), x^k - x^*\rangle + \alpha_k^2 \|g(x^k;\xi^k)\|^2 \end{aligned} \tag{5.23}$$

We deal with each term in this expansion separately. We take the expectation of both sides with respect to all the random variables encountered by the algorithm up to and including iteration $k$, namely $i_0, i_1, \ldots, i_k$. By applying the law of iterated expectation, and noting that $x^k$ depends on $\xi^0, \xi^1, \ldots, \xi^{k-1}$ *but not on* $\xi^k$, we obtain

$$\begin{aligned}\mathbb{E}[\langle g(x^k;\xi^k), x^k - x^*\rangle] &= \mathbb{E}\left[\mathbb{E}_{\xi^k}[\langle g(x^k;\xi^k), x^k - x^*\rangle \mid \xi^0, \xi^1, \ldots, \xi^{k-1}]\right] \\ &= \mathbb{E}\left[\langle \mathbb{E}_{\xi^k}[g(x^k;\xi^k) \mid \xi^0, \xi^1, \ldots, \xi^{k-1}], x^k - x^*\rangle\right] \\ &= \mathbb{E}\left[\langle \nabla f(x^k), x^k - x^*\rangle\right] .\end{aligned}$$

In the last step of this derivation, we used the fact that $g(x^k;\xi^k)$ depends on $\xi^k$, while $x^k$ does not, so we took the expectation of $g(x^k;\xi^k)$ explicitly with respect to $\xi^k$, to obtain $\nabla f(x^k)$.

By a similar argument, we can bound the last term in (5.23) by using (5.19):

$$\mathbb{E}[\|g(x^k;\xi^k)\|_2^2] = \mathbb{E}\left[\mathbb{E}_{\xi^k}[\|g(x^k;\xi^k)\|_2^2 \mid \xi^0, \xi^1, \ldots, \xi^{k-1}]\right] \leq \mathbb{E}[L_g^2 \|x^k - x^*\|_2^2 + B^2].$$

By defining the squared expected error as

$$A_k := \mathbb{E}[\|x^k - x^*\|^2], \tag{5.24}$$

we obtain by taking expectations of both sides of (5.23) and substituting these relationships that

$$A_{k+1} \leq (1 + \alpha_k^2 L_g^2) A_k - 2\alpha_k \mathbb{E}\left[\langle \nabla f(x^k), x^k - x^*\rangle\right] + \alpha_k^2 B^2 . \tag{5.25}$$

Our results follow from different manipulations of (5.25) for different settings of $L_g$ and $B$. We proceed through several cases.

### 5.4.1 Case 1: $L_g = 0$.

When $L_g = 0$, the expression (5.25) reduces to

$$A_{k+1} \leq A_k - 2\alpha_k \mathbb{E}\left[\langle \nabla f(x^k), x^k - x^* \rangle\right] + \alpha_k^2 B^2 \,. \tag{5.26}$$

Define $\lambda_k$ to be the sum of all stepsizes up to and including iteration $k$, and $\bar{x}^k$ to be the average of all iterates so far, weighted by the stepsizes $\alpha_j$, that is,

$$\lambda_k = \sum_{j=0}^{k} \alpha_j, \quad \bar{x}^k = \lambda_k^{-1} \sum_{j=0}^{k} \alpha_j x^j \,. \tag{5.27}$$

(We made use of averaged iterates in the analysis of mirror descent also, in Section 3.7.) We analyze the deviation of $f(\bar{x}_k)$ from optimality. Given the initial point $x^0$, which we assume to now be random, we define $D_0 := \|x^0 - x^*\|$ to be the initial squared error. (Note from (5.24) that $A_0 = D_0^2$.) After $T$ iterations, we have the following estimate for $\bar{x}^T$:

$$\begin{aligned}
\mathbb{E}[f(\bar{x}^T) - f(x^*)] &\leq \mathbb{E}\left[\lambda_T^{-1} \sum_{j=0}^{T} \alpha_j (f(x^j) - f(x^*))\right] && \text{(5.28a)} \\
&\leq \lambda_T^{-1} \sum_{j=0}^{T} \alpha_j \mathbb{E}[\langle \nabla f(x^j), x^j - x^* \rangle] && \text{(5.28b)} \\
&\leq \lambda_T^{-1} \sum_{j=0}^{T} \left[\tfrac{1}{2}(A_j - A_{j+1}) + \tfrac{1}{2}\alpha_j^2 B^2\right] && \text{(5.28c)} \\
&= \frac{1}{2}\lambda_T^{-1}\left[A_0 - A_{T+1} + B^2 \sum_{j=0}^{T} \alpha_j^2\right] \\
&\leq \frac{D_0^2 + B^2 \sum_{j=0}^{T} \alpha_j^2}{2\sum_{j=0}^{T} \alpha_j}. && \text{(5.28d)}
\end{aligned}$$

Here, (5.28a) follows from convexity of $f$ and the definition of $\bar{x}^T$; (5.28b) again uses convexity of $f$; and (5.28c) follows from (5.26).

With the bound (5.28d) in hand, we can prove the following result for the case of constant stepsizes: $\alpha_k \equiv \alpha > 0$ for all $k$.

**Proposition 5.1** ([68]). *Suppose we run SG on a convex $f$ with $L_g = 0$ for $T$ steps with constant stepsize $\alpha > 0$. Define*

$$\alpha_{\text{opt}} = \frac{D_0}{B\sqrt{T+1}} \quad \textit{and} \quad \theta := \frac{\alpha}{\alpha_{\text{opt}}}.$$

*Then we have the bound*

$$\mathbb{E}[f(\bar{x}^T) - f^\star] \leq \left(\tfrac{1}{2}\theta + \tfrac{1}{2}\theta^{-1}\right) \frac{BD_0}{\sqrt{T+1}} \,. \tag{5.29}$$

*Proof.* The proof comes directly from setting $\alpha_j \equiv \alpha = \theta\alpha_{\rm opt} = \theta\frac{D_0}{B\sqrt{T+1}}$ in (5.28d). We have

$$\mathbb{E}\left[f\left(\bar{x}^T\right) - f(x_\star)\right] \leq \frac{D_0^2 + B^2(T+1)\alpha^2}{2(T+1)\alpha} = \left(\tfrac{1}{2}\theta^{-1} + \tfrac{1}{2}\theta\right)\frac{BD_0}{\sqrt{T+1}}\,.$$

□

The tightest bound is attained when $\theta = 1$, that is, $\alpha = \alpha_{\rm opt}$. The bound degrades approximately linearly in the error factor in our choice of $\alpha$. That is, if our $\alpha$ differs by a factor of 2 (in either direction) from $\alpha_{\rm opt}$, the bound is worse by a factor of approximately 2. This means that to achieve the same bound as with the optimal step size, we need to take about *four times* as many iterations, because the bound also depends on the iteration counter $T$ through a factor of approximately $1/\sqrt{T}$.

Other stepsize schemes could also be selected here, including choices of $\alpha_k$ that decrease with $k$. But the constant stepsize is optimal for an upper bound of this type.

### 5.4.2 Case 2: $B = 0$

When $B = 0$, we obtain a *linear* rate of convergence in the expected-error measure $A_k$. The expression (5.25) simplifies in this case to

$$A_{k+1} \leq (1 + \alpha_k^2 L_g^2)A_k - 2\alpha_k \mathbb{E}\left[\langle \nabla f(x^k), x^k - x^*\rangle\right]. \tag{5.30}$$

Supposing that $f$ is strongly convex, with modulus of convexity $m > 0$, we have that

$$\langle \nabla f(x), x - x^\star\rangle \geq m\|x - x^\star\|^2\,. \tag{5.31}$$

By substituting into (5.30), we obtain

$$A_{k+1} \leq (1 - 2m\alpha_k + L_g^2\alpha_k^2)A_k\,. \tag{5.32}$$

By choosing a constant steplength $\alpha_k \equiv \alpha$, for any $\alpha$ in the range $(0, 2m/L_g^2)$, we obtain a linear rate of convergence. The optimal choice of $\alpha$ is the one that minimizes the factor $(1 - 2m\alpha + L_g^2\alpha^2)$ in the right-hand side of (5.32), that is, $\alpha = m/L_g^2$. For this choice, we obtain from (5.32) that $A_{k+1} \leq (1 - m^2/L_g^2)A_k$, $k = 0, 1, 2, \ldots$, so that

$$A_k \leq \left(1 - \frac{m^2}{L_g^2}\right)^k D_0^2. \tag{5.33}$$

We can use this expression to bound the number of iterations $T$ required to guarantee that the expected error $\mathbb{E}\left[\|x^T - x^\star\|^2\right] = A_T$ falls below a specified threshold $\epsilon > 0$. By applying the technique in Section A.2 to (5.33) we find that

$$T = \left\lceil \frac{L_g^2}{m^2}\log\left(\frac{D_0^2}{\epsilon}\right)\right\rceil.$$

**Special case: The Kaczmarz method.** For problems with additional structure, we can obtain even faster rates of convergence. In particular, faster rates are achievable for the randomized Kaczmarz method where we specialize our analysis to overdetermined least squares problems (5.15). Recall that we assume that each vector $a_i$ has unit norm and that there exists an $x^*$ (possibly nonunique) such that $a_i^T x^* = b_i$ for all $i$. Consider the stochastic gradient method with stepsize 1:

$$x^{k+1} = x^k - a_{i_k}(a_{i_k}^T x^k - b_{i_k})$$

where $i_k$ is chosen uniformly at random each iteration. We have

$$\begin{aligned}\|x^{k+1} - x^*\|^2 &= \left\|x^k - a_{i_k}(a_{i_k}^T x^k - b_{i_k}) - x^*\right\|^2 \\ &= \|x^k - x^*\|^2 - 2\left(a_{i_k}^T(x^k - x^*)(a_{i_k}^T x^k - b_{i_k})\right) + (a_{i_k}^T x^k - b_{i_k})^2 \\ &= \|x^k - x^*\|^2 - (a_{i_k}^T x^k - b_{i_k})^2\,,\end{aligned}$$

where we used $a_{i_k}^T(x^k - x^*) = a_{i_k}^T x^k - b_{i_k}$. Let $A$ be the matrix whose rows are the vectors $a_i$, and let $\lambda_{\min,\mathrm{nz}}$ denote the minimum nonzero eigenvalue of $A^T A$. It can be shown that by choosing $x^*$ to be the point that minimizes $\|x - x^*\|$, among all points satisfying $Ax^* = b$, that $\|Ax - b\|^2 \geq \lambda_{\min,\mathrm{nz}}\|x - x^*\|^2$ (see Section A.7). By taking expectations, we obtain for this value of $x^*$ that

$$\begin{aligned}\mathbb{E}\left[\|x^{k+1} - x^*\|^2 | x^k\right] &\leq \|x^k - x^*\|^2 - \mathbb{E}_{i_k}\left[(a_{i_k}^T x^k - b_{i_k})^2\right] \\ &= \|x^k - x^*\|^2 - \frac{1}{n}\|Ax^k - b\|^2 \\ &\leq \left(1 - \frac{\lambda_{\min,\mathrm{nz}}}{n}\right)\|x^k - x^*\|^2.\end{aligned}$$

Defining $D_k := \min_{x^* \,:\, Ax^* = b} \|x^k - x^*\|^2$, we have from $D_{k+1} \leq \|x^{k+1} - x^*\|^2$ and $D_k = \|x^k - x^*\|^2$ (because of the way that $x^*$ is defined above) that

$$\mathbb{E}\, D_{k+1} \leq \mathbb{E}\,\|x^{k+1} - x^*\|^2 \leq \left(1 - \frac{\lambda_{\min,\mathrm{nz}}}{n}\right)\mathbb{E}\, D_k.$$

This is a faster rate of convergence than the one we derived for the general case where $B = 0$.

### 5.4.3 Case 3: $B$ and $L_g$ both nonzero

In the general case in which both $B$ and $L_g$ are nonzero, but $f$ is strongly convex, we have by using (5.31) in (5.25) that

$$A_{k+1} \leq (1 - 2m\alpha_k + \alpha_k^2 L_g^2)A_k + \alpha_k^2 B^2 \tag{5.34}$$

**Constant Stepsize.** First, consider the case of a constant stepsize. Assuming that $\alpha \in (0, 2m/L_g^2)$, we can roll out the recursion (5.34) to obtain

$$A_k \leq (1 - 2m\alpha + \alpha^2 L_g^2)^k D_0^2 + \frac{\alpha B^2}{2m - \alpha L_g^2}\,. \tag{5.35}$$

No matter how many iterations $k$ are taken, the bound on the right-hand size never falls below the threshold value

$$\frac{\alpha B^2}{2m - \alpha L_g^2}\,. \tag{5.36}$$

This behavior can be observed in practice. The iterates converge to a ball around the optimal solution, whose radius is bounded by (5.36), but from that point forward, they bounce around inside this ball. We can reduce the radius of the ball by decreasing $\alpha$, but this has the effect of slowing the linear rate of convergence indicated by the first term in the right-hand side of (5.35): The quantity $1 - 2m\alpha + \alpha^2 L_g^2$ moves closer to 1.

One way to balance these two effects is to make use of epochs, as we discuss below in Section 5.5.1.

**Diminishing Stepsize.** The scheme just described suggests another approach, one in which we decrease the stepsize $\alpha_k$ at a rate approximately proportional to $1/k$. (The epoch-doubling scheme is a piece-constant approximation to this. At the last iterate of epoch $S$, we will have taken about $(2^S - 1)T$ total iterations, and the current steplength will be $\alpha/2^{S-1}$.)

Suppose we choose the stepsize to satisfy

$$\alpha_k = \frac{\gamma}{k_0 + k},$$

where $\gamma$ and $k_0$ are constants to be determined. We will show that suitable choices of these constants lead to an error bound of the form

$$A_k \leq \frac{Q}{k_0 + k}\,,$$

for some $Q$. The following proposition can be proved by induction.

**Proposition 5.2.** *Suppose $f$ is strongly convex with modulus of convexity $m$. If we run SG with stepsize*

$$\alpha_k = \frac{1}{2m(L_g^2/2m^2 + k)}, \quad k = 0, 1, 2, \ldots,$$

*then we have for some numerical constant $c_0$,*

$$\mathbb{E}[\|x^k - x^\star\|^2] \leq \frac{c_0 B^2}{2m(L_g^2/2m^2 + k)}, \quad k = 0, 1, 2, \ldots\,.$$

## 5.5 Implementation Aspects

We mention here several techniques that are important elements of many practical implementations of SG.

### 5.5.1 Epochs

As mentioned in Section 5.4.3, *epochs* are a central concept in SG. In an epoch, some number of iterations are run, and then a choice is made about whether to change the stepsize. A common strategy is to run with a constant step size for some fixed number of iterations $T$, and then reduce

the stepsize by a constant factor $\gamma$. Thus, if our initial stepsize is $\alpha$, on the $k$th epoch, the stepsize is $\alpha\gamma^{k-1}$. This method is often more robust in practice than the diminishing stepsize rule. For this stepsize rule, a reasonable heuristic is to choose $\gamma$ between 0.8 and 0.9. (Tuning of these "hyperparameters" is one of the most important issues in practical implementation of SG.)

Another popular rule is called *epoch doubling*. In this scheme, we run for $T$ steps with stepsize $\alpha$, then run $2T$ steps with stepsize $\alpha/2$, and then $4T$ steps with stepsize $\alpha/4$ and so on. Note that this scheme provides a piecewise constant approximation to the function $\alpha/k$.

### 5.5.2 Minibatching

When applying SG to the finite-sum objective (5.5), steps are often not based just on the gradient on a single term in this sum but rather on a *minibatch* of terms, usually of a given size (say p). That is, at iteration $k$, we select a subset $\mathcal{S}_k \subset \{1, 2, \ldots, n\}$ with $|\mathcal{S}_k| = p$, and set

$$x^{k+1} = x^k - \alpha_k \frac{1}{p} \sum_{i \in \mathcal{S}_k} \nabla f_i(x^k).$$

If the subset $\mathcal{S}_k$ is chosen uniformly at random among the set of all subsets of size $p$ from $\{1, 2, \ldots, n\}$, and is i.i.d. across the iterations $k$, the convergence theory outlined above can be applied. The idea is that the minibatch has lower variance as an estimate of $\nabla f(x^k)$ than does the estimate based on a single term, namely, $\nabla f_{i_k}(x^k)$, so more rapid convergence can be expected. Of course, it is also typically $p$ times more expensive to obtain this estimate! Still, when we account for the cost of performing the update to the vector $x$ and possibly communicating this update to the nodes in a parallel processing architecture, the minibatching approach makes sense. It is used almost universally in practical implementations of SG. The choice of minibatch size $p$ is another "hyperparameter" that can influence significantly the practical performance of the approach.

### 5.5.3 Momentum

A popular variant of SG makes use of *momentum*, replacing the basic step (5.2) with one of the form

$$x^{k+1} = x^k - \alpha_k g(x^k, \xi^k) + \beta_k (x^k - x^{k-1}). \tag{5.37}$$

The inspiration for this approach comes, of course, from the accelerated gradient methods of Chapter 4. In practice, these variants are highly successful, with popular choices for $\beta_k$ often falling in the range $[.8, .95]$.

In the case when $B = 0$, as in the randomized Kaczmarz method, the use of momentum can yield speedups comparable to those seen in the accelerated gradient methods of Chapter 4. The overhead of computing and maintaining the momentum term can cancel out the gains in speedup. (See further discussion in the Notes and References.)

In the general case, the theoretical guarantees for momentum methods demonstrate only meager gains over standard SG. Essentially, we know that the function value will converge at a rate of $1/k$, but for certain instances, one can reduce the constant in front of the $1/k$ using momentum or acceleration. Regardless of the theoretical guarantees, one should always keep in mind that momentum can provide significant practical accelerations, and it should be considered an option in any implementation of SG.

## Notes and References

The foundational paper for SG is Robbins and Monro [84]. As we mentioned, similar ideas were proposed independently in other contexts. Among these we can count Rosenblatt's perceptron algorithm [89], discussed in Section 5.1.3. Application of SG to problems in machine learning were described by first by Zhang in 2004 [112], and later by the authors of the Pegasos paper [90], who described a minibatched SG approach for linear SVM.

Analysis of SG for the case of $L_g = 0$, for both weakly and strongly convex cases, appears in [68]. (This paper did much to popularize the SG approach in the optimization community.)

The Kaczmarz algorithm dates to 1937 [50]. It was used as the standard method in image reconstruction from tomographic data for many years. The description of a randomized variant by Strohmer and Vershynin in 2009 [94] generated a new wave of interest in the approach, leading to the development of many new variants with interesting properties.

The ideas behind empirical risk minimization for learning were described by Vapnik in a 1992 confernece paper [103] and in his classic text [104].

Incremental gradient was described in the context of least squares by Bertsekas [8], who also wrote a survey (in a more general context) in [10]. Another interesting contribution on this topic is [13].

The past few years have seen a number of principled approaches emerge for using acceleration in conjunction with stochastic gradient. Accelerated SG has been described for least squares in [49]. A general (but complex) approach called Katyusha, due to Allen-Zhu, is described in [1]. (An convergence analysis of Katyusha and other SG methods based on dissipativity theory and semindefinite programs appears in [47].)

## Exercises

1. Consider the $k$th iteration (5.12) of the cyclic incremental gradient method applied to the function (5.11). Show that the minimizer is found exactly after $N$ steps (that is $x^N = x^*$) and that $f(x^*)$ is one-half of the variance of the set $\{\omega_1, \omega_2, \ldots, \omega_N\}$.

2. Verify the formula (5.14), given that the mean of the random variable $\omega$ is $\mu$ and its variance is $\sigma^2$. (The random variables $\omega_i$, $i = 1, 2, \ldots, k$ follow the same distribution, and all the random variables in this expression are independent.)

3. We showed that the unregularized support vector machine (5.21) admits a bound of the form (5.19) with $L_g = 0$. Find values of $L_g$ and $B$ such that the *regularized* support vector machine (5.7) (with $g(w^k, \xi^k)$ defined by (5.8)) satisfies (5.19). (Hint: Use the inequality $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$.)

4. (a) Consider the finite-sum objective (5.5) with additive Gaussian noise model on the component functions $f_i$, that is,

$$[\nabla f_i(x)]_j = [\nabla f(x)]_j + \epsilon_{ij}, \quad \text{for all } i = 1, 2, \ldots, N \text{ and } j = 1, 2, \ldots, n,$$

where $\epsilon_{ij} \sim N(0, \sigma^2)$ for all $i, j$. Show that when we estimate the gradient using a

minibatch $\mathcal{S} \subset \{1, 2, \ldots, N\}$, that is,

$$g = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \nabla f_i(x),$$

then we have

$$\mathbb{E}(\|g - \nabla f(x)\|^2) = \frac{n}{|\mathcal{S}|}\sigma^2, \quad \mathbb{E}(\|g\|^2) = \|\nabla f(x)\|^2 + \frac{n}{|\mathcal{S}|}\sigma^2.$$

(b) Consider a minibatch strategy for the additive Gaussian noise model (5.3) for the general formulation (5.1). That is, the gradient estimate is

$$g(x; \xi_1, \xi_2, \ldots, \xi_s) := \nabla f(x) + \frac{1}{s}\sum_{j=1}^{s} \xi_j,$$

where each $\xi_j$ is i.i.d. with distribution $N(0, \sigma^2 I)$, and $s \geq 1$. Show that

$$\mathbb{E}_{\xi_1, \xi_2, \ldots, \xi_s}\left(\|g(x; \xi_1, \xi_2, \ldots, \xi_s)\|^2\right) = \|\nabla f(x)\|^2 + \frac{d}{s}\sigma^2.$$

5. A popular heuristic in training neural networks is called *dropout*. Suppose we are running stochastic gradient descent on a function on $\mathbb{R}^n$. In each iteration of stochastic gradient descent, a subset $S \subset \{1, 2, \ldots, n\}$ of variables is chosen at random. A stochastic gradient is computed with those coordinates in $S$ set to 0. Then, only the coordinates in the complement $S^c$ are updated. Suppose we are minimizing the least squares cost

$$f(x) = \frac{1}{2N}\sum_{i=1}^{N} (a_i^T x - b_i)^2 .$$

Find a function $\hat{f}(x)$ such that each iteration of dropout SGD corresponds to taking a valid step of the incremental gradient method applied to $\hat{f}$. Qualitatively, how does changing the cardinality $S$ change the solution to which dropout SGD converges?

6. Let $f(x) = \mathbb{E}[F(x; \xi)]$ be a strongly convex function with parameter $m$. Assume that

$$\mathbb{E}[\|\nabla F(x; \xi)\|^2] \leq L_g^2 \|x - x^*\|^2 + B^2 ,$$

where $x^*$ denotes the minimizer of $f$ and $L_g$ and $B$ are constants. Suppose we run the stochastic gradient method on $f$ by sampling $\xi$ and taking steps along $\nabla F(x; \xi)$ using an epoch doubling approach. That is, we run for $T$ steps with stepsize $\alpha$, and then $2T$ steps with stepsize $\alpha/2$, and then $4T$ steps with stepsize $\alpha/4$ and so on. Let $\hat{x}_t$ be the average of all of the iterates in the $t$th epoch. How many epochs are required to guarantee that $\mathbb{E}\left[\|\hat{x}_t - x^*\|^2\right] \leq \epsilon$?

7. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a strongly convex function with $L$-Lipschitz gradients and strong convexity parameter $m$. Consider an algorithm that performs exact line searches along random search directions. Each iterate uses the following scheme to move from current iterate $x$ to the next iterate $x^+$.

(a) Choose a direction $v$ randomly from $N(0, \sigma^2 I)$ (independently of the search directions at all previous iterations);

(b) Set $t_{\min} = \arg\min_t f(x + tv)$;

(c) Set $x^+ = x + t_{\min} v$.

Prove that $\mathbb{E}[f(x^T) - f(x^*)] \leq \epsilon$ provided

$$T \geq \frac{CnL}{m} \log\left(\frac{f(x^0) - f(x^*)}{\epsilon}\right),$$

for some constant $C$. What is the most appropriate value for $C$? (Hints: Use Lemma 2.2 to deduce that

$$f(x + tv) \leq f(x) + tv^T \nabla f(x) + \frac{L}{2} t^2 \|v\|^2.$$

Use that if $v \sim N(0, \sigma^2 I)$, then for any component $v_j$ of $v$, $j = 1, 2, \ldots, n$, we have $\mathbb{E}_v\, v_j^2 / \|v\|^2 = 1/n$. Also use the bound (3.10).)

8. Consider applying stochastic gradient with *constant stepsize* $\alpha \in (0, 1)$ to (5.11), so that each iteration has the form

$$x^{k+1} = x^k - \alpha(x^k - \omega_{i_k})$$

for $i_k$ drawn uniformly at random from $\{1, 2, \ldots, N\}$. Assuming that the initial point is $x^0 = 0$, write down an explicit expression for $x^k$, and find $\mathbb{E}_{i_0, i_1, \ldots, i_{k-1}}(x^k)$.

9. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex, differentiable function and let $g(x, \xi)$ be a continuous function satisfying (5.1), where $\xi$ is a random variable from set $\Xi$ with distribution $P$. Consider a projected SG approach on a compact convex set $\Omega$, whose iterates are defined as follows

$$x^{k+1} = P_\Omega\left(x^k - \alpha g(x^k, \xi^k)\right), \quad k = 0, 1, 2, \ldots,$$

where $\xi^k$ is chosen randomly with distribution $P$ and $\alpha$ is a constant step size. Defining $\bar{x}^T := \sum_{t=0}^T x^t/(T+1)$ (consistently with (5.27)), prove that this algorithm converges at the following rate:

$$\mathbb{E}\, f(\bar{x}^T) - \min_{x \in \Omega} f(x) \leq \frac{c}{\sqrt{T+1}},$$

where $c$ is a problem-specific constant.

10. Consider the convex quadratic function defined in (5.11). where $x \in \mathbb{R}^n$ and $\omega_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, N$. The vectors $\omega_i$ have the following additional properties:

$$\sum_{i=1}^N \omega_i = 0, \quad \|\omega_i\| = 1, \text{ for all } i = 1, 2, \ldots, N.$$

Consider the SG iteration defined by $x^{k+1} = x^k - \alpha_k(x^k - \omega_{i_k})$, where $i_k$ is selected i.i.d. uniformly from $\{1, 2, \ldots, N\}$, for some steplength $\alpha_k > 0$, where $x^0$ is any initial point.

(a) Show that the minimizer of $f$ is $x^* = 0$.

(b) Express the conditional expectation $\mathbb{E}_{i_k}(\|x^{k+1}\|^2 \,|\, x^k)$ in terms of $\|x^k\|^2$ and $\alpha_k$.

(c) By applying the bound in (b) recursively, and using the notation $A_K := \mathbb{E}(\|x^K\|^2)$, find a bound for $e_K$ for any $K = 1, 2, \ldots$ in terms of $e_0 = \|x^0\|^2$ and $\alpha_0, \alpha_1, \ldots, \alpha_{K-1}$, where $\mathbb{E}$ denotes the expectation with respect to all random variables $i_0, i_1, i_2, \ldots$. (Hint: Derive the formula for the first few values of $K$, that is, $A_1, A_2, A_3 \ldots$, until you see the patterm emerge.)

(d) Simplify the bound in (c) for the case in which all steplengths are the same, that is, $\alpha_k \equiv \alpha$ for all $k = 0, 1, 2, \ldots$.

(e) Do you expect the iterates $\{x^k\}$ generated from the constant-steplength variant in (d) to converge to the solution $x^* = 0$? Do you expect them to converge to a *ball* around the solution? If so, what is the approximate radius of this ball?

(f) Consider choosing the steplengths $\alpha_k = 1/(k+2)$, $k = 0, 1, 2, \ldots$. From your answer in part (c), can you say that $\mathbb{E}(\|x^K\|^2) \to 0$ as $K \to \infty$ for this choice of steplengths? Explain.