14

Stochastic Gradient Methods for Principled Estimation with Large Data Sets

Edo Airoldi and Panos Toulis

CONTENTS

14.1 Introduction

Parameter estimation by optimization of an objective function, such as maximum-likelihood and maximum a posteriori, is a fundamental idea in statistics and machine learning (Fisher, 1922, Lehmann and Casella, 2003, Hastie et al., 2011). However, widely used optimization-based estimation algorithms, such as Fisher scoring, the Expectation-Maximization (EM) algorithm, and iteratively reweighted least squares (Fisher, 1925, Dempster et al., 1977, Green, 1984), are not scalable to modern data sets with hundreds of millions of data points and hundreds or thousands of covariates (National Research Council, 2013).

To illustrate, let us consider the problem of estimating the true parameter value $\theta_\star \in \mathbb{R}^p$ from an i.i.d. sample $D = \{X_n, Y_n\}$, for $n = 1, 2, \ldots, N$; $X_n \in \mathbb{R}^p$ is the covariate vector, and $Y_n \in \mathbb{R}^d$ is the outcome distributed conditionally on X_n according to the known

distribution f and unknown model parameters $\theta_\star$,

$$Y_n|X_n \sim f(\cdot; X_n, \theta_\star)$$

We assume that the data points (X_n, Y_n) are observed in sequence (streaming data). The log-likelihood, $\log f(Y; X, \theta)$, as a function of the parameter value θ given a data point (X, Y), will be denoted by $\ell(\theta; Y, X)$; for brevity, we define $\ell(\theta; D) = \sum_{n=1}^{N} \ell(\theta; X_n, Y_n)$ as the complete data log-likelihood.

Traditional estimation methods are typically iterative and have a running time complexity that ranges between $O(Np^3)$ and $O(Np)$ in worst cases and best cases, respectively. Newton–Raphson methods, for instance, update an estimate θ_{n-1} of the parameters through the recursion

$$\theta_n = \theta_{n-1} - H_{n-1}^{-1} \nabla \ell(\theta_{n-1}; D) \tag{14.1}$$

where $H_n = \nabla\nabla\ell(\theta_n; D)$ is the $p \times p$ Hessian matrix of the complete data log-likelihood. The matrix inversion and the likelihood computation over the data set D imply complexity $O(Np^{2+\epsilon})$, which makes the algorithm unsuited for estimation with large data sets. Fisher scoring replaces the Hessian matrix in Equation 14.1 with its expected value over a data point (X_n, Y_n), that is, it uses the Fisher information matrix $\mathcal{I}(\theta) = -\mathbb{E}\left(\nabla\nabla\ell(\theta; X_n, Y_n)\right)$. The advantage of this method is that a steady increase in the likelihood is possible because the difference

$$\ell(\theta + \epsilon\Delta\theta; D) - \ell(\theta; D) \approx \epsilon\, \ell(\theta; D)^\intercal \mathcal{I}(\theta)^{-1} \ell(\theta; D) + O(\epsilon^2)$$

can be made positive for an appropriately small value $\epsilon > 0$, because $\mathcal{I}(\theta)$ is positive definite. However, Fisher scoring is computationally comparable to Newton–Raphson's, and thus it is also unsuited for estimation with large data sets. Other general estimation algorithms, such as EM or iteratively reweighted least squares (Green, 1984), have similar computational constraints.

Quasi-Newton methods are a powerful alternative that is widely used in practice. In quasi-Newton methods, the Hessian in the Newton–Raphson algorithm is approximated by a low-rank matrix that is updated at each iteration as new values of the gradient become available. This yields an algorithm with complexity $O(Np^2)$, or $O(Np)$ in certain favorable cases (Hennig and Kiefel, 2013).

However, estimation with large data sets requires complexity that scales linearly with N, the number of data points, but sublinearly with p, the number of parameters. The first requirement on N seems hard to overcome because each data point carries information for $\theta_\star$ by the i.i.d. data assumption. Therefore, gracious scaling with p is necessary.

Such computational requirements have recently sparked interest in procedures that utilize only *first-order* information, that is, methods that utilize only the gradient function. A prominent procedure that fits this description is *stochastic gradient descent* (SGD), defined through the iteration

$$\theta_n^{\text{sgd}} = \theta_{n-1}^{\text{sgd}} + a_n \nabla \ell(\theta_{n-1}^{\text{sgd}}; X_n, Y_n) \tag{14.2}$$

We will refer to procedure 14.2 as SGD with *explicit updates*, or *explicit SGD* for short, because the next iterate θ_n^{sgd} can be computed immediately after the new data point (X_n, Y_n) is observed. The sequence $a_n > 0$ is known as the *learning rate* sequence, typically defined such that $na_n \to \alpha > 0$, as $n \to \infty$. The parameter $\alpha > 0$ is the *learning rate parameter*, and it is crucial for the convergence and stability of explicit SGD.

From a computational perspective, the SGD procedure (Equation 14.2) is appealing because the expensive inversion of $p \times p$ matrices, as in Newton–Raphson, is replaced by a

single sequence of scalars $a_n > 0$. Furthermore, the log-likelihood is evaluated at a single data point (X_n, Y_n), and not on the entire data set D.

From a theoretical perspective, the explicit SGD procedure is justified because Equation 14.2 is a special case of the stochastic approximation method of Robbins and Monro (1951). By the theory of stochastic approximation, explicit SGD converges to a point θ_∞ that satisfies $\mathbb{E}\left(\nabla\ell(\theta_\infty; X, Y)\right) = 0$; under typical regularity conditions, θ_∞ is exactly the true parameter value $\theta_\star$. As a recursive statistical estimation method, explicit SGD was first proposed by Sakrison (1965) in a simple second-order form, that is, using the Fisher information matrix in iteration (Equation 14.2); the simplicity of SGD has also made it very popular in optimization and machine learning with large data sets (Le Cun and Bottou, 2004, Zhang, 2004, Spall, 2005).*

However, the remarkable simplicity of explicit SGD comes at a price, as the SGD procedure requires careful tuning of the learning rate parameter α. For small values of α, the iterates θ_n^{sgd} will converge very slowly to $\theta_\star$ (large bias), whereas for large values of α, the iterates θ_n^{sgd} will either have a large asymptotic variance (with respect to random data D), or even diverge numerically. In large data sets with many parameters (large p), the balance between bias, variance, and stability is very delicate, and nearly impossible to achieve without appropriately modifying Equation 14.2.

Interestingly, the simple modification of explicit SGD defined through the iteration

$$\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + a_n \nabla\ell(\theta_n^{\text{im}}; X_n, Y_n) \tag{14.3}$$

can resolve its stability issue virtually at no cost. We will refer to procedure 14.3 as *implicit stochastic gradient descent*, or *implicit SGD* for short (Toulis et al., 2014, Toulis and Airoldi, 2015); Equation 14.3 is implicit because the next iterate θ_n^{im} appears on both sides of the equation. This equation is a p-dimensional fixed-point equation, which is generally hard to solve. However, for a large family of statistical models, it can be reduced to a one-dimensional fixed-point equation; we discuss computational issues of implicit SGD in Section 14.3.4.

The first intuition for implicit SGD is obtained using a Taylor expansion of the implicit update (Equation 14.3). In particular, assuming a common point $\theta_{n-1}^{\text{sgd}} = \theta_{n-1}^{\text{im}} = \theta$, a Taylor expansion of Equation 14.3 around θ_{n-1}^{im} implies

$$\Delta\theta_n^{\text{im}} = \left(\mathbb{I} + a_n \hat{\mathcal{I}}(\theta; X_n, Y_n)\right)^{-1} \Delta\theta_n^{\text{sgd}} + O(a_n^2) \tag{14.4}$$

where:

$\Delta\theta_n = \theta_n - \theta_{n-1}$ for both explicit and implicit methods

$\hat{\mathcal{I}}(\theta; X_n, Y_n) = -\nabla\nabla\ell(\theta; X_n, Y_n)$ is the observed Fisher information matrix

$\mathbb{I}$ is the $p \times p$ identity matrix

Thus, implicit SGD uses updates that are a *shrinked* version of the explicit ones; the shrinkage factor in Equation 14.4 depends on the observed information up to the nth data point, and is similar to shrinkage in ridge regression.

Naturally, implicit SGD has also a Bayesian interpretation. In particular, if the log-likelihood is continuously differentiable, then the update in Equation 14.3 is equivalent to the update

$$\theta_n^{\text{im}} = \arg\max_{\theta \in \mathbb{R}^p} \left\{ a_n \ell(\theta; X_n, Y_n) - \frac{1}{2}||\theta - \theta_{n-1}^{\text{im}}||^2 \right\} \tag{14.5}$$

*Recursive estimation methods using stochastic approximation were originally developed for problems with streaming data. However, these methods are more broadly applicable to estimation with a static data set. Asymptotically, these two scenarios are equivalent, with the estimates converging to the true parameter value $\theta_\star$. Estimates with a static data set (that is, with a finite sample) converge instead to the point that minimizes some predefined empirical loss, for example, based on the likelihood.

The iterate θ_n^{im} from Equation 14.5 is the posterior mode of the following Bayesian model:

$$\begin{aligned} \theta|\theta_{n-1}^{\text{im}} &\sim \mathcal{N}(\theta_{n-1}^{\text{im}}, a_n\mathbb{I}) \\ Y_n|X_n,\theta &\sim f(\cdot;X_n,\theta) \end{aligned} \tag{14.6}$$

where $\mathcal{N}$ denotes the normal distribution. Therefore, the learning rate a_n relates to the information received after n data points have been observed, and encodes our trust on the current estimate θ_{n-1}^{im}. The Bayesian formulation (Equation 14.6) demonstrates the flexibility of implicit SGD. For example, depending on the parameter space of $\theta_\star$, the Bayesian model in Equation 14.6 could be different; for instance, if $\theta_\star$ was a scale parameter, then the normal distribution could be replaced by an inverse chi-squared distribution. Furthermore, instead of $a_n\mathbb{I}$ as the prior variance, it would be statistically efficient to use the Fisher information matrix $(1/n)\mathcal{I}(\theta_{n-1}^{\text{im}})^{-1}$, completely analogous to Sakrison's method—we discuss these ideas in Section 14.3.5.1.

There is also a tight connection of Equation 14.5 to *proximal methods* in optimization. For example, if we replaced the stochastic component $\ell(\theta;X_n,Y_n)$ in Equation 14.5 with the complete data log-likelihood $\ell(\theta;D)$, then procedure 14.5 would be essentially the proximal point algorithm of Rockafellar (1976) that applies to deterministic settings. This algorithm is known for its numerical stability, and has been generalized through the idea of splitting algorithms (Lions and Mercier, 1979); see Parikh and Boyd (2013) for a comprehensive review. The convergence of proximal methods with a stochastic component, as in Equation 14.5, has been analyzed recently—under various forms and assumptions—by Bertsekas (2011), Ryu and Boyd (2014), and Rosasco et al. (2014). From a statistical perspective, Toulis and Airoldi (2014) derived the asymptotic variance of θ_n^{sgd} and θ_n^{im} as estimators of $\theta_\star$, and provided an algorithm to efficiently compute Equation 14.5 for the family of generalized linear models—we show a generalization of this result in Section 14.3.4. In the online learning literature, regret analyses of implicit methods have been given by Kivinen et al. (2006) and Kulis and Bartlett (2010). Further intuitions for proximal methods (Equation 14.5) have been given by Krakowski et al. (2007) and Nemirovski et al. (2009), who showed that proximal methods can fit better in the geometry of the parameter space.

Arguably, the normalized least mean squares (NLMS) filter (Nagumo and Noda, 1967) was the first statistical model that used an implicit update as in Equation 14.3, and was shown to be robust to input noise (Slock, 1993). Two other recent stochastic proximal methods are Prox-SVRG (Xiao and Zhang, 2014) and Prox-SAG (Schmidt et al., 2013, section 6). The main idea in both methods is to replace the gradient in Equation 14.5 with an estimate of the full gradient averaged over all data points that has the same expectation with the gradient of Equation 14.3 but smaller variance. Because of their operational complexity, we will not discuss these methods further. Instead, in Section 14.3.5.1, we will discuss a related proximal method, namely AdaGrad (Duchi et al., 2011a), that maintains one learning rate for each parameter component, and updates these learning rates as new data points are observed.

Example 14.1 *Consider the linear normal model, $Y_n|X_n \sim \mathcal{N}(X_n^\intercal\theta_\star, 1)$. The log-likelihood for this model is $\ell(\theta;X_n,Y_n) = -\frac{1}{2}(Y_n - X_n^\intercal\theta_\star)^2$. Therefore, the explicit SGD procedure will be*

$$\theta_n^{\text{sgd}} = \theta_{n-1}^{\text{sgd}} + a_n(Y_n - X_n^\intercal\theta_{n-1}^{\text{sgd}})X_n = (\mathbb{I} - a_nX_nX_n^\intercal)\theta_{n-1}^{\text{sgd}} + a_nY_nX_n \tag{14.7}$$

Equation 14.7 is known as the least mean squares filter (LMS) in signal processing, or as the Widrow–Hoff algorithm (Widrow and Hoff, 1960), and it is a special case of explicit SGD.

The stability problems of explicit SGD become apparent by inspection of Equation 14.7; a misspecification of a_n can lead to a poor next iterate θ_n^{sgd}, for example, when $\mathbb{I} - a_n X_n X_n^\intercal$ has large negative eigenvalues—we discuss these issues in Section 14.3.2.

The implicit SGD procedure for the linear model is

$$\begin{aligned}\theta_n^{\text{im}} &= \theta_{n-1}^{\text{im}} + a_n(Y_n - X_n^\intercal \theta_n^{\text{im}})X_n \Rightarrow \\ \theta_n^{\text{im}} &= (\mathbb{I} + a_n X_n X_n^\intercal)^{-1}\theta_{n-1}^{\text{im}} + a_n(\mathbb{I} + a_n X_n X_n^\intercal)^{-1} Y_n X_n \end{aligned} \tag{14.8}$$

Equation 14.8 is known as the NLMS filter in signal processing (Nagumo and Noda, 1967). In contrast to explicit SGD, the implicit iterate θ_n^{im} is a weighted average between the previous iterate θ_{n-1}^{im} and the new observation $Y_n X_n$, which is now stable to misspecifications of the learning rate a_n.

14.1.1 Outline

The structure of this chapter is as follows. In Section 14.2, we give an overview of the Robbins–Monro procedure and Sakrison's recursive estimation method, which provide the theoretical basis for SGD methods. In Section 14.3, we introduce a simple generalization of explicit and implicit SGD, and we analyze them as *statistical estimation procedures* for the model parameters $\theta_\star$ after n data points have been observed. In Section 14.3.1, we give results on the frequentist statistical properties of SGD estimators, that is, their asymptotic bias and variance across multiple realizations of the data set D. We then leverage these results to study optimal learning rate sequences a_n (Section 14.3.3), the loss of statistical information in SGD, and numerical stability (Section 14.3.2). In Section 14.3.5, we illustrate three extensions of the SGD methods, in particular: (1) second-order SGD methods (Section 14.3.5.1), which adaptively approximate the Fisher information matrix; (2) averaged SGD methods, which use larger learning rates together with averaging of the iterates; and (3) Monte Carlo SGD methods, which can be applied when the log-likelihood cannot be efficiently computed. In Section 14.4, we review applications of SGD in statistics and machine learning, namely, online EM, Markov Chain Monte Carlo (MCMC) posterior sampling, reinforcement learning, and deep learning.

14.2 Stochastic Approximation

Consider a random variable $H(\theta)$ that depends on parameter θ; for simplicity, assume that $H(\theta)$ and θ are real numbers. The regression function, $h(\theta) = \mathbb{E}\,(H(\theta))$, is decreasing but possibly unknown. Robbins and Monro (1951) considered the problem of finding the unique point $\theta_\star$ for which $h(\theta_\star) = 0$. They devised a procedure, known as *the Robbins–Monro procedure*, in which an estimate θ_{n-1} of $\theta_\star$ is utilized to sample one new data point $H(\theta_{n-1})$; by definition, $\mathbb{E}\,(H(\theta_{n-1})|\,\theta_{n-1}) = h(\theta_{n-1})$. The estimate is then updated according to the following rule:

$$\theta_n = \theta_{n-1} + a_n H(\theta_{n-1}) \tag{14.9}$$

The scalar $a_n > 0$ is the *learning rate* and should decay to zero, but not too fast to guarantee convergence. Robbins and Monro (1951) proved that $\mathbb{E}\left((\theta_n - \theta_\star)^2\right) \to 0$ if,

(a) $\mathbb{E}\left(H(\theta)^2 \middle|\, \theta\right) < \infty$, for any θ, and

(b) $\sum_{i=1}^{\infty} a_i = \infty$ and $\sum_{i=1}^{\infty} a_i^2 < \infty$.

Extensions to multiple dimensions were soon given by Blum (1954). The necessary conditions for convergence in such cases are the negative definiteness of the Jacobian of h, or that H is the stochastic gradient of a function with unique zero (Wei, 1987, Ruppert, 1988b, section 4).

The original proof of Robbins and Monro (1951) is technical but the main idea is straightforward. Let $b_n \triangleq \mathbb{E}\left((\theta_n - \theta_\star)^2\right)$ denote the squared error of the iterates in Equation 14.9; then from iteration (Equation 14.9), it follows:

$$b_n = b_{n-1} + 2a_n\mathbb{E}\left((\theta_{n-1} - \theta_\star)h(\theta_{n-1})\right) + a_n^2\mathbb{E}\left(H(\theta_{n-1})^2\right)$$

In the neighborhood of $\theta_\star$, we assume that $h(\theta_{n-1}) \approx h'(\theta_\star)(\theta_{n-1} - \theta_\star)$, and thus

$$b_n = (1 + 2a_n h'(\theta_\star))b_{n-1} + a_n^2\mathbb{E}\left(H(\theta_{n-1})^2\right) \tag{14.10}$$

For a learning rate $a_n = \alpha/n$, using typical techniques in stochastic approximation (Chung, 1954), we can derive from Equation 14.10 that $b_n \to 0$. Furthermore, $nb_n \to \alpha^2\sigma^2(2\alpha|h'(\theta_\star)| - 1)^{-1}$, where $\sigma^2 \triangleq \mathbb{E}\left(H(\theta_\star)^2\right)$, as shown by several authors (Chung, 1954, Sacks, 1958, Fabian, 1968). Clearly, the learning rate parameter α is critical for the performance of the Robbins–Monro procedure. Its optimal value is $\alpha_\star = 1/h'(\theta_\star)$, which requires knowledge of the true parameter value $\theta_\star$, and the slope of h at that point. This optimality result inspired an important line of research on *adaptive* stochastic approximation methods, such as the Venter process (Venter, 1967), in which quantities that are important for the convergence and efficiency of iterates θ_n (for example, the quantity $h'(\theta_\star)$) are being estimated as the stochastic approximation proceeds.

14.2.1 Sakrison's recursive estimation method

Although initially motivated by sequential experiment design, the Robbins–Monro procedure was soon modified for statistical estimation. Similar to the estimation setup in Section 14.1, Sakrison (1965) was interested in estimating the parameters $\theta_\star$ of a model that generated i.i.d. observations (X_n, Y_n), in a way that is computationally and statistically efficient. Sakrison first recognized that one could set $H(\theta) \triangleq \nabla \log \ell(\theta; X_n, Y_n)$ in the Robbins–Monro procedure (Equation 14.9), and use the identity $\mathbb{E}\left(\nabla\ell(\theta_\star; X_n, Y_n)\right) = 0$ to show why the procedure will converge to the true parameter value $\theta_\star$. Sakrison's *recursive estimation method* was essentially the first *explicit* SGD method proposed in the literature:

$$\theta_n = \theta_{n-1} + a_n\mathcal{I}(\theta_{n-1})^{-1}\nabla\ell(\theta_{n-1}; X_n, Y_n) \tag{14.11}$$

where a_n is a learning rate sequence that satisfies the Robbins–Monro conditions of Section 14.2. The SGD procedure 14.11 is *second order* because it uses the Fisher information matrix in addition to the log-likelihood gradient. By the theory of stochastic approximation, $\theta_n \to \theta_\star$, and thus $\mathcal{I}(\theta_n) \to \mathcal{I}(\theta_\star)$. Sakrison (1965) proved that $n\mathbb{E}\left(||\theta_n - \theta_\star||^2\right) \to \text{trace}(\mathcal{I}(\theta_\star)^{-1})$, which indicates that estimation of $\theta_\star$ is asymptotically optimal, that is, it achieves the minimum variance of the maximum-likelihood estimator. However, Sakrison's method is not computationally efficient, as it requires an expensive matrix inversion at every iteration. Still, it reveals that the estimation of the Fisher information matrix is essential for optimal SGD. Adaptive second-order methods leverage this insight to approximate the Fisher information matrix, and improve upon first-order SGD methods.

14.3 Estimation with Stochastic Gradient Methods

We slightly generalize the SGD methods in Section 14.1 through the definitions

$$\theta_n^{\text{sgd}} = \theta_{n-1}^{\text{sgd}} + a_n C \nabla \ell(\theta_{n-1}^{\text{sgd}}; X_n, Y_n) \tag{14.12}$$

$$\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + a_n C \nabla \ell(\theta_n^{\text{im}}; X_n, Y_n) \tag{14.13}$$

where C is symmetric and positive definite, and commutes with $\mathcal{I}(\theta_\star)$; adaptive second-order methods where C is updated at every iteration are discussed in Section 14.3.5.1. The iterate θ_n^{sgd} is the explicit SGD estimator of $\theta_\star$ after the nth data point has been observed; similarly, θ_n^{im} is the implicit SGD estimator of $\theta_\star$. The total number of data points, denoted by N, will be assumed to be practically infinite. We will then compare the asymptotic variance of those estimators with the variance of the maximum-likelihood estimator on n data points, which, under typical regularity conditions, has variance $\frac{1}{n}\mathcal{I}(\theta_\star)^{-1}$. The evaluation is done from a frequentist perspective, that is, across multiple realizations of the data set up to n data points $D = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$, under the same model f and true parameter value $\theta_\star$.*

Typically, both SGD methods have two phases, namely the *exploration phase* and the *convergence phase* (Amari, 1998, Bottou and Murata, 2002). In the exploration phase, the iterates approach $\theta_\star$, whereas in the convergence phase, they jitter around $\theta_\star$ within a ball of slowly decreasing radius. We will overview a typical analysis of SGD in the final convergence phase, where a Taylor approximation in the neighborhood of $\theta_\star$ is assumed accurate (Murata, 1998, Toulis et al., 2014). In particular, let $\mu(\theta) = \mathbb{E}\left(\nabla \ell(\theta; X_n, Y_n)\right)$, and assume

$$\mu(\theta_n) = \mu(\theta_\star) + J_\mu(\theta_\star)(\theta_n - \theta_\star) + o(a_n) \tag{14.14}$$

where:

J_μ is the Jacobian of the function $\mu(\cdot)$

$o(a_n)$ denotes a vector sequence with norms of order $o(a_n)$

Under typical regularity conditions, $\mu(\theta_\star) = 0$ and $J_\mu(\theta_\star) = -\mathcal{I}(\theta_\star)$ (Lehmann and Casella, 1998).

14.3.1 Bias and variance

Denote the biases of the two SGD methods with $\mathrm{E}(\theta_n^{\text{sgd}} - \theta_\star) \triangleq b_n^{\text{sgd}}$ and $\mathrm{E}(\theta_n^{\text{im}} - \theta_\star) \triangleq b_n^{\text{im}}$. Then, by taking expectations in Equations 14.12 and 14.13 we obtain the recursions

$$b_n^{\text{sgd}} = \left(\mathbb{I} - a_n C \mathcal{I}(\theta_\star)\right) b_{n-1}^{\text{sgd}} + o(a_n) \tag{14.15}$$

$$b_n^{\text{im}} = \left(\mathbb{I} + a_n C \mathcal{I}(\theta_\star)\right)^{-1} b_{n-1}^{\text{im}} + o(a_n) \tag{14.16}$$

We observe that convergence—the rate at which the two methods become unbiased in the limit—differs in the two SGD methods. The explicit SGD method converges faster than the implicit one because $||(\mathbb{I} - a_n C \mathcal{I}(\theta_\star))|| < ||(\mathbb{I} + a_n C \mathcal{I}(\theta_\star))^{-1}||$, for sufficiently large n, but the rates become equal in the limit as $a_n \to 0$. However, the implicit method compensates by being more stable in the specification of the learning rate sequence and the

*This is an important distinction because, traditionally, the focus in optimization has been to obtain fast convergence to a parameter value that minimizes the empirical loss, for example, the maximum-likelihood. From a statistical viewpoint, under variability of the data, there is a tradeoff between convergence to an estimator and the estimator's asymptotic variance (Le Cun and Bottou, 2004).

condition matrix C. Loosely speaking, the bias b_n^{im} cannot be much worse than b_{n-1}^{im} because $(\mathbb{I} + a_n C\mathcal{I}(\theta_\star))^{-1}$ is a contraction matrix, for any choice of $a_n > 0$. Exact nonasymptotic derivations for the bias of explicit SGD are given by Moulines and Bach (2011), and for the bias of implicit SGD by Toulis and Airoldi (2014).

Regarding statistical efficiency, Toulis et al. (2014) showed that, if $(2C\mathcal{I}(\theta_\star) - \mathbb{I}/\alpha)$ is positive definite, it holds that

$$n\text{Var}(\theta_n^{\text{sgd}}) \to \alpha^2(2\alpha C\mathcal{I}(\theta_\star) - \mathbb{I})^{-1}C\mathcal{I}(\theta_\star)C^\intercal$$
$$n\text{Var}(\theta_n^{\text{im}}) \to \alpha^2(2\alpha C\mathcal{I}(\theta_\star) - \mathbb{I})^{-1}C\mathcal{I}(\theta_\star)C^\intercal \quad (14.17)$$

where $\alpha = \lim_{n\to\infty} na_n$ is the learning rate parameter of SGD, as defined in Section 14.1. Therefore, both SGD methods have the same asymptotic efficiency, which depends on the learning rate parameter α and the Fisher information matrix $\mathcal{I}(\theta_\star)$. Intuitively, the term $(2\alpha C\mathcal{I}(\theta_\star) - \mathbb{I})^{-1}$ in Equation 14.17 is a factor that shows how much information is lost by the SGD methods. For example, setting $C = \mathcal{I}(\theta_\star)^{-1}$ and $\alpha = 1$, implies $(2\alpha C\mathcal{I}(\theta_\star) - \mathbb{I})^{-1} = \mathbb{I}$, and the asymptotic variance for both estimators is $(1/n)\mathcal{I}(\theta_\star)^{-1}$, that is, it is the minimum variance attainable by the maximum-likelihood estimator. This is exactly Sakrison's result presented in Section 14.2.1.

Asymptotic variance results similar to Equation 14.17, but not in the context of model estimation, were first studied in the stochastic approximation literature by Chung (1954), Sacks (1958), and followed by Fabian (1968) and several other authors (see also Ljung et al., 1992, parts I, II), where more general formulas are possible using a Lyapunov equation.

14.3.2 Stability

Stability has been a well-known issue for explicit SGD (Gardner, 1984, Amari et al., 1997). In practice, the main problem is that the learning rate sequence a_n needs to agree with the eigenvalues of the Fisher information matrix $\mathcal{I}(\theta_\star)$. To see this, let us simplify Equations 14.15 and 14.16 by dropping the remainder terms $o(a_n)$. It follows that

$$b_n^{\text{sgd}} = (\mathbb{I} - a_n C\mathcal{I}(\theta_\star))b_{n-1}^{\text{sgd}} = P_1^n b_0$$
$$b_n^{\text{im}} = (\mathbb{I} + a_n C\mathcal{I}(\theta_\star))^{-1}b_{n-1}^{\text{im}} = Q_1^n b_0 \quad (14.18)$$

where $P_1^n = \prod_{i=1}^n(\mathbb{I} - a_i C\mathcal{I}(\theta_\star))$, $Q_1^n = \prod_{i=1}^n(\mathbb{I} + a_i C\mathcal{I}(\theta_\star))^{-1}$, and b_0 denotes the initial bias of the two procedures from a common starting point θ_0. The matrices P_1^n and Q_1^n describe how fast the initial bias b_0 decays for both SGD methods. For small-to-moderate n, the two matrices critically affect the stability of SGD methods. For simplicity, we compare those matrices assuming rate $a_n = \alpha/n$ and a fixed condition matrix $C = \mathbb{I}$.

Under such assumptions, the eigenvalues of P_1^n can be calculated as $\lambda_i' = \prod_{j=1}^n(1 - \alpha\lambda_i/j) = O(n^{-\alpha\lambda_i})$, for $0 < \alpha\lambda_i < 1$, where λ_i are the eigenvalues of the Fisher information matrix $\mathcal{I}(\theta_\star)$. Thus, the magnitude of P_1^n will be dominated by $\lambda_{\max}$, the maximum eigenvalue of $\mathcal{I}(\theta_\star)$, and the rate of convergence to zero will be dominated by $\lambda_{\min}$, the minimum eigenvalue of $\mathcal{I}(\theta_\star)$. For stable eigenvalues λ_i', the terms in the aforementioned product need to be less than 1; therefore, it is desirable that $|1 - \alpha\lambda_{\max}| \leq 1 \Rightarrow \alpha \leq 2/\lambda_{\max}$. For statistical efficiency, it is desirable that $(2\alpha\mathcal{I}(\theta_\star) - \mathbb{I})$ is positive definite, as shown in Equation 14.17, and so $\alpha > 1/(2\lambda_{\min})$. In high-dimensional settings, the conditions for stability and efficiency are hard to satisfy simultaneously because $\lambda_{\max}$ is usually much larger than $\lambda_{\min}$. Thus, in explicit SGD, a small learning rate can guarantee stability, but this comes at a price in convergence, which will be at the order of $O(n^{-\alpha\lambda_{\min}})$. On the other hand, a large learning rate increases the convergence rate but it comes at a price in stability.

In stark contrast, the implicit procedure is *unconditionally stable*. The eigenvalues of Q_1^n are $\lambda_i' = \prod_{j=1}^n 1/(1+\alpha\lambda_i/j) = O(n^{-\alpha\lambda_i})$, and thus are guaranteed to be less than 1 for any choice of the learning rate parameter α, because $(1+\alpha\lambda_i/j)^{-1} < 1$, for every i and $\alpha > 0$. The critical difference with explicit SGD is that it is no longer required to have a small α for stability because the eigenvalues of Q_1^n are always less than 1.

Based on this analysis, the magnitude of P_1^n can become arbitrarily large, and thus explicit SGD is likely to numerically diverge. In contrast, Q_1^n is guaranteed to be bounded and so, under any misspecification of the learning rate parameter, implicit SGD is guaranteed to remain bounded. The instability of explicit SGD is well known and requires careful work to be avoided in practice. In the following section, we focus on the related task of selecting the learning rate sequence.

14.3.3 Choice of learning rate sequence

An interesting observation about the asymptotic variance results (Equation 14.17) is that, for any choice of the learning rate parameter α, it holds that

$$\alpha^2(2\alpha C\mathcal{I}(\theta_\star) - I)^{-1} C\mathcal{I}(\theta_\star)C^\intercal \geq \mathcal{I}(\theta_\star)^{-1} \tag{14.19}$$

where $A \geq B$ indicates that $A - B$ is nonnegative definite for two matrices A and B. Hence, both SGD methods incur an information loss when compared to the maximum-likelihood estimator, and the loss can be quantified exactly through Equation 14.17. Such information loss can be avoided if we set $C = \mathcal{I}(\theta_\star)^{-1}$ and $\alpha = 1$.* However, this requires knowledge of the Fisher information matrix on the true parameters $\theta_\star$, which are unknown. The Venter process (Venter, 1967) was the first method to follow an adaptive approach to estimate the Fisher matrix, and was later analyzed and extended by several other authors (Fabian, 1973, Lai and Robbins, 1979, Amari et al., 2000, Bottou and Le Cun, 2005). Adaptive methods that perform an approximation of the matrix $\mathcal{I}(\theta_\star)$ (for example, through a quasi-Newton scheme) have recently been applied with considerable success (Schraudolph et al., 2007, Bordes et al., 2009); we review such methods in Section 14.3.5.1.

However, an efficiency loss is generally unavoidable in first-order SGD, that is, when $C = \mathbb{I}$. In such cases, there is no loss only when the eigenvalues λ_i of the Fisher information matrix are identical. When those eigenvalues are distinct, one reasonable way to set the learning rate parameter α is to minimize the trace of the asymptotic variance matrix in Equation 14.17, that is, solve

$$\hat{\alpha} = \arg\min_\alpha \sum_i \frac{\alpha^2\lambda_i}{(2\alpha\lambda_i - 1)} \tag{14.20}$$

under the constraint that $\alpha > 1/(2\lambda_{\min})$, thus making an undesirable but necessary compromise for convergence in all parameter components. However, the eigenvalues λ_i are unknown in practice and need to be estimated from the data. This problem has received significant attention recently and several methods exist (see Karoui, 2008, and references within).

Several more options for setting the learning rate are available, due to a voluminous amount of research literature on learning rate sequences for stochastic approximation and SGD. In general, the learning rate for explicit SGD should be of the form $a_n = \alpha(\alpha\beta + n)^{-1}$. Parameter α controls the asymptotic variance (see Equation 14.17), and a reasonable choice

*Equivalently, we could have a sequence of matrices C_n that converges to $\mathcal{I}(\theta_\star)^{-1}$, as in Sakrison's procedure (Sakrison, 1965).

is the solution of Equation 14.20, which requires estimates of the eigenvalues of the Fisher information matrix $\mathcal{I}(\theta_\star)$. A simpler choice is to use $\alpha = 1/\lambda_{\min}$, where $\lambda_{\min}$ is the minimum eigenvalue of $\mathcal{I}(\theta_\star)$; the value $1/\lambda_{\min}$ is an approximate solution of Equation 14.20 with good empirical performance (Xu, 2011, Toulis et al., 2014). Parameter β is used to stabilize explicit SGD. In particular, it normalizes the learning rate to account for the variance of the stochastic gradient $\text{Var}\left(\nabla\ell(\theta_n; X_n, Y_n)\right) = \mathcal{I}(\theta_\star) + O(a_n)$, for points near $\theta_\star$. One reasonable value is $\beta = \text{trace}(\mathcal{I}(\theta_\star))$, which can be estimated easily by summing norms of the score function, that is, $\hat{\beta} = \sum_{i=1}^{n} ||\nabla\ell(\theta_{i-1}; X_i, Y_i)||^2$. This idea is extended to multiple dimensions by Amari et al. (2000), Duchi et al. (2011b) and Schaul et al. (2012); we discuss further in Section 14.3.5.1.

For implicit SGD, a learning rate sequence $a_n = \alpha(\alpha+n)^{-1}$ works well in practice (Toulis et al., 2014). As before, α controls statistical efficiency, and we can set $\alpha = 1/\lambda_{\min}$, as in explicit SGD. The additional stability term β of explicit SGD is unnecessary in implicit SGD because the implicit method performs an indirect normalization of the learning rate—this is similar to shrinkage described in Equation 14.4.

Eventually, tuning the learning rate sequence depends on problem-specific considerations, and there is a considerable variety of sequences that have been employed in practice (George and Powell, 2006, Schaul et al., 2012). Principled design of learning rates in first-order SGD remains an important research topic; for example, recent work has investigated variance reduction techniques (Johnson and Zhang, 2013, Wang et al., 2013), or even constant learning rates for least-squares models (Bach and Moulines, 2013). Second-order methods that essentially maintain multiple learning rates, one for each parameter component, are discussed in Section 14.3.5.1.

14.3.4 Efficient computation of implicit methods

The update in implicit SGD (Equation 14.3) is a p-dimensional fixed-point equation, which is generally hard to solve. However, in many statistical models, Equation 14.3 can be reduced to a one-dimensional fixed-point equation, which can be computed very fast using a numerical root-finding method.

Consider a linear statistical model where $\ell(\theta; X_n, Y_n)$ depends on θ only through the linear term $X_n^\intercal\theta$. A large family of models satisfy this condition: generalized linear models, generalized additive models, proportional hazards, etc. We denote $\ell(\theta; X_n, Y_n) = g_n(X_n^\intercal\theta)$, where we suppressed the dependence of g on X_n, Y_n in the subscript n. Then, $\nabla\ell(\theta; X_n, Y_n) = g_n'(X_n^\intercal\theta)X_n$, and therefore the direction of the gradient of the log-likelihood is parameter free. It follows that the implicit procedure can be written as

$$\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + a_n\lambda_n\nabla\ell(\theta_{n-1}^{\text{im}}; X_n, Y_n) \tag{14.21}$$

where the gradient is now calculated at the previous estimate θ_{n-1}^{im}, and λ_n is an appropriate scaling. We now derive λ_n by combining the definition of implicit SGD (Equations 14.3 and 14.21):

$$\begin{aligned}\theta_{n-1}^{\text{im}} + a_n\lambda_n\nabla\ell(\theta_{n-1}^{\text{im}}; X_n, Y_n) = \theta_{n-1}^{\text{im}} + a_n\nabla\ell(\theta_n^{\text{im}}; X_n, Y_n) \Rightarrow \\ \lambda_n g_n'(X_n^\intercal\theta_{n-1}^{\text{im}}) = g_n'(X_n^\intercal\theta_n^{\text{im}})\end{aligned} \tag{14.22}$$

Using Equation 14.21 in 14.22, we get

$$\lambda_n = \frac{g_n'\left(X_n^\intercal\theta_{n-1}^{\text{im}} + a_n\lambda_n||X_n||^2 g_n'(X_n^\intercal\theta_{n-1}^{\text{im}})\right)}{g_n'(X_n^\intercal\theta_{n-1}^{\text{im}})} \tag{14.23}$$

Equation 14.23 is a one-dimensional fixed-point equation with respect to λ_n. Thus, the implicit iterate θ_n^{im} of Equation 14.3 can be efficiently computed by first obtaining λ_n from

Equation 14.23, and then using Equation 14.21. Narrow search bounds for Equation 14.23 are usually available; see, for example, algorithm 1 by Toulis et al. (2014) for implicit SGD on generalized linear models, and algorithm 2 by Tran et al. (2015b) for implicit SGD on the Cox proportional hazards model. Fast implementation of implicit SGD methods are included in the `sgd` R package (Tran et al., 2015a,b).

14.3.5 Extensions

Here, we illustrate three extensions of the SGD methods: second-order SGD methods, which adaptively approximate the Fisher information matrix; averaged SGD methods, which use larger learning rates together with averaging of the iterates; and Monte Carlo SGD, which can be applied when the log-likelihood cannot be efficiently computed.

14.3.5.1 Second-order methods

Sakrison's recursive estimation method (Equation 14.11) is the archetype of second-order SGD, but it requires prior knowledge of the Fisher information matrix $\mathcal{I}(\theta_\star)$. Several methods aim to recursively estimate the Fisher information matrix, and use those estimates within the main procedure of estimating $\theta_\star$; such methods are known as *adaptive*. Early adaptive methods in stochastic approximation were given by Nevelson and Khasminskiĭ (1973), Wei (1987), and Spall (2000); translated into an SGD procedure, such methods recursively estimate $\mathcal{I}(\theta_\star)$ by fixing a covariate value X_n, perturbing the parameter estimate θ_{n-1}—for example, by taking $\theta_{n-1} \pm \epsilon u$, where $\epsilon > 0$ is a small constant and u is a basis vector—and then sampling outcome Y_n, given the fixed covariate and parameter values. While such methods are very useful when one has control over the data generation process as, for example, in sequential experiment design, they are impractical for modern estimation tasks with large data sets.

A simple and effective approach to recursively estimate $\mathcal{I}(\theta_\star)$ was developed by Amari et al. (2000). The idea is to estimate $\mathcal{I}(\theta_\star)$ through a separate stochastic approximation procedure, and use the estimate $\hat{\mathcal{I}}$ in the procedure for $\theta_\star$ as follows:

$$\begin{aligned}\hat{\mathcal{I}}_n &= (1 - c_n)\hat{\mathcal{I}}_{n-1} + c_n \nabla \ell(\theta_{n-1}; X_n, Y_n) \nabla \ell(\theta_{n-1}; X_n, Y_n)^\intercal \\ \theta_n &= \theta_{n-1} + a_n \hat{\mathcal{I}}_n^{-1} \nabla \ell(\theta_{n-1}; X_n, Y_n)\end{aligned} \tag{14.24}$$

Inversion of the estimate $\hat{\mathcal{I}}_n$ is relatively cheap through the Sherman–Morrison formula. This scheme, however, introduces the additional problem of determining the sequence c_n in Equation 14.24. Amari et al. (2000) advocated for a small constant $c_n = c > 0$ determined through computer simulations.

An alternative approach based on quasi-Newton methods was developed by Bordes et al. (2009). Their method, termed *SGD-QN*, approximated the Fisher information matrix through a *secant condition* as in the original BFGS algorithm (Broyden, 1965). The secant condition in SGD-QN is

$$\theta_n - \theta_{n-1} \approx \hat{\mathcal{I}}_{n-1}^{-1} \left[\nabla \ell(\theta_n; X_n, Y_n) - \nabla \ell(\theta_{n-1}; X_n, Y_n)\right] \triangleq \hat{\mathcal{I}}_{n-1}^{-1} \Delta \ell_n \tag{14.25}$$

where $\hat{\mathcal{I}}_n$ are kept diagonal. If L_n denotes the diagonal matrix with the ith diagonal element $L_{ii} = (\theta_{n,i} - \theta_{n-1,i})/\Delta \ell_{n,i}$, then the update for $\hat{\mathcal{I}}_n$ is

$$\hat{\mathcal{I}}_n \leftarrow \hat{\mathcal{I}}_{n-1} + \frac{2}{r}(L_n - \hat{\mathcal{I}}_{n-1}) \tag{14.26}$$

whereas the update for θ_n is similar to Equation 14.24. The parameter r is controlled internally in the algorithm, and counts the number of times the update (Equation 14.26) has been performed.

Another notable second-order method is *AdaGrad* (Duchi et al., 2011b), which maintains multiple learning rates using gradient information only. In one popular variant of the method, AdaGrad keeps a $p \times p$ diagonal matrix of learning rates A_n that is updated at every iteration; on observing data (X_n, Y_n), AdaGrad updates A_n as follows:

$$A_n = A_{n-1} + \text{diag}(\nabla \ell(\theta_{n-1}; X_n, Y_n) \nabla \ell(\theta_{n-1}; X_n, Y_n)^\intercal) \tag{14.27}$$

where $\text{diag}(\cdot)$ is the diagonal matrix with the same diagonal as its matrix argument. Estimation with AdaGrad proceeds through the iteration

$$\theta_n = \theta_{n-1} + \alpha A_n^{-1/2} \nabla \ell(\theta_{n-1}; X_n, Y_n) \tag{14.28}$$

where $\alpha > 0$ is shared among all parameter components. The original motivation for AdaGrad stems from proximal methods in optimization, but there is a statistical intuition why the update (Equation 14.28) is reasonable. In many dimensions, where some parameter components affect outcomes less frequently than others, AdaGrad *estimates* the information that has *actually* been received for each component. A conservative estimate of this information is provided by the elements of A_n in Equation 14.27, which is justified because, under typical conditions, $\mathbb{E}\left(\nabla \ell(\theta; X_n, Y_n) \nabla \ell(\theta; X_n, Y_n)^\intercal\right) = \mathcal{I}(\theta)$.

All the second-order methods presented so far are explicit; however, they can have straightforward variants using implicit updates. For example, in the method of Amari et al. (2000), one can use the implicit update

$$\theta_n = \theta_{n-1} + a_n \hat{\mathcal{I}}_n^{-1} \nabla \ell(\theta_n; X_n, Y_n) \tag{14.29}$$

instead of the explicit one in Equation 14.24. The solution of Equation 14.29 does not present additional challenges, compared to Section 14.3.4, because inverses of the estimates $\hat{\mathcal{I}}_n$ are easy to compute.

14.3.5.2 Averaging

In certain models, second-order methods can be avoided and still statistical efficiency can be achieved through a combination of larger learning rates a_n with averaging of the iterates θ_n. The corresponding SGD procedure is usually referred to as *averaged SGD*, or *ASGD* for short.* Averaging in stochastic approximation was studied independently by Ruppert (1988a) and Bather (1989), who proposed similar averaging schemes. If we use the notation of Section 14.2, Ruppert (1988a) considered the following modification of the Robbins–Monro procedure (Equation 14.9):

$$\begin{aligned} \theta_n &= \theta_{n-1} + a_n H(\theta_{n-1}) \\ \bar{\theta}_n &= \frac{1}{n} \sum_{i=1}^{n} \theta_i \end{aligned} \tag{14.30}$$

where $a_n = \alpha n^{-c}$, $1/2 < c < 1$, and $\bar{\theta}_n$ are considered the estimates of $\theta_\star$, instead of θ_n. Under certain conditions, Ruppert (1988a) showed that $n\text{Var}(\bar{\theta}_n) \to \sigma^2 / h'(\theta_\star)^2$, where $\sigma^2 = \text{Var}\left(H(\theta) | \, \theta = \theta_\star\right)$. Therefore, $\bar{\theta}_n$ achieves the minimum variance that is possible according to the analysis in Section 14.2.

Ruppert (1988a) gives a nice statistical intuition on why averaging with larger learning rates implies statistical efficiency. First, write $H(\theta_n) = h(\theta_n) + \varepsilon_n$, where ε_n are

*The acronym ASGD is also used in machine learning to denote *asynchronous* SGD, that is, a variant of SGD that can be parallelized on multiple machines. We will not consider this variant here.

zero-mean independent random variables with finite variance. By solving the recursion 14.9, we get

$$\theta_n - \theta_\star = \sum_{i=1}^{n} \gamma_{in} a_i \varepsilon_i + o(1) \tag{14.31}$$

where $\gamma_{in} = \exp\{-A(n) + A(i)\}$, $A(m) = K \sum_{j=1}^{m} a_j$ is the function of partial sums, and K is a constant. Ruppert (1988a) shows that Equation 14.31 can be rewritten as

$$\theta_n - \theta_\star = a_n \sum_{i=b(n)}^{n} \gamma_{in} \varepsilon_i + o(1) \tag{14.32}$$

where $b(n) = \lfloor n - n^c \log n \rfloor$, and $\lfloor \cdot \rfloor$ is the positive integer floor function. When $a_n = a/n$, Ruppert (1988a) shows that $b(n) = O(1)$ and $\theta_n - \theta_\star$ is the weighted average over all noise variables ε_n. In this case, there is significant autocorrelation in the series θ_n, and averaging actually can make things worse. However, when $a_n = \alpha n^{-c}$, for $1/2 < c < 1$, $\theta_n - \theta_\star$ is a weighted average of only $O(n^c \log n)$ noise variables. In this case, the iterates $\theta_{\lfloor p_1 n \rfloor}$ and $\theta_{\lfloor p_2 n \rfloor}$, for $0 < p_1 < p_2 < 1$, are asymptotically uncorrelated, and thus averaging improves estimation efficiency.

Polyak and Juditsky (1992) derive further significant results for averaged SGD, showing in particular that ASGD can be asymptotically efficient as second-order SGD under certain conditions (for example, strong convexity of the expected log-likelihood). In fact, ASGD is usually referred to as the *Polyak–Ruppert averaging scheme*. Adoption of averaging schemes for statistical estimation has been slow but steady over the years (Zhang, 2004, Nemirovski et al., 2009, Bottou, 2010, Cappé, 2011). One practical reason is that a bad selection of the learning rate sequence can cause ASGD to converge more slowly than classical explicit SGD (Xu, 2011). Such problems can be avoided by using implicit SGD with averaging because implicit methods can afford larger learning rates that can speed up convergence. At the same time using implicit updates in procedure 14.30 still maintains asymptotic efficiency (Toulis et al., 2015).

14.3.5.3 Monte Carlo stochastic gradient descent

A key requirement for the application of SGD procedures is that the likelihood is easy to evaluate. However, this is not possible in many situations, for example, when the likelihood is only known up to a normalizing constant. In such cases, definitions 14.12 and 14.13 cannot be applied directly because $\nabla \ell(\theta; X, Y) = S(X, Y) - Z(\theta)$, and while S is easy to compute, Z is hard to compute as it entails a multidimensional integral.

However, if sampling from the model is feasible, then a variant of explicit SGD, termed *Monte Carlo SGD* (Toulis and Airoldi, 2014), can be constructed to take advantage of the identity $\mathbb{E}\,(\nabla \ell(\theta_\star; X, Y)) = 0$, which implies $\mathbb{E}\,(S(X, Y)) = Z(\theta_\star)$. Starting from an estimate θ_0^{mc}, we iterate the following steps for $n = 1, 2, \ldots$:

1. Observe covariate X_n and outcome Y_n; compute $S_n \triangleq S(X_n, Y_n)$.
2. Get m samples $\widetilde{Y}_{n,i} | X_n, \theta_{n-1}^{\text{mc}} \sim f(\cdot; X_n, \theta_{n-1}^{\text{mc}})$, for $i = 1, 2, \ldots, m$.
3. Compute statistic $\tilde{S}_{n-1} \triangleq (1/m) \sum_{i=1}^{m} S(X_n, \widetilde{Y}_{n,i})$.
4. Update estimate θ_{n-1}^{mc} through

$$\theta_n^{\text{mc}} = \theta_{n-1}^{\text{mc}} + a_n C (S_n - \widetilde{S}_{n-1}) \tag{14.33}$$

This method is valid under typical assumptions of stochastic approximation theory because it converges to a point $\theta_\infty^{\text{mc}}$ such that $\mathbb{E}\left(S(X,\widetilde{Y})|\theta_\infty^{\text{mc}}\right) = \mathbb{E}\left(S(X,Y)\right) = Z(\theta_\star)$, and thus $\mathbb{E}\left(\nabla\ell(\theta_\infty^{\text{mc}}, X, Y)\right) = 0$ as required. Furthermore, the asymptotic variance of estimates of Monte Carlo SGD satisfies

$$n\text{Var}\left(\theta_n^{\text{mc}}\right) \to (1+1/m)\cdot\alpha^2(2\alpha C\mathcal{I}(\theta_\star) - I)^{-1}C\mathcal{I}(\theta_\star)C^\intercal \tag{14.34}$$

which exceeds the variance of the typical explicit (or implicit) SGD estimator in Equation 14.17 by a factor of $(1+1/m)$.

In its current form, Monte Carlo SGD (Equation 14.33) is only explicit; an implicit version would require to sample data from the next iterate, which is technically challenging. Still, an *approximate* implicit implementation of Monte Carlo SGD is possible through shrinkage, for example, through shrinking θ_n^{mc} by a factor $(\mathbb{I} + a_n\mathcal{I}(\theta_n^{\text{mc}}))^{-1}$, or more easily by $(1 + a_n\text{trace}(\mathcal{I}(\theta_n^{\text{mc}})))^{-1}$.

Theoretically, Monte Carlo SGD is based on *sampling-controlled* stochastic approximation methods (Dupuis and Simha, 1991), in which the usual regression function of the Robbins–Monro procedure (Equation 14.9) is only accessible through sampling, for example, through MCMC. Convergence in such settings is subtle because it depends on the ergodicity of the underlying Markov chain (Younes, 1999). Finally, when perfect sampling from the underlying model is not possible, we may use samples $\tilde{S}_n$ that are obtained by a handful of MCMC steps, even before the chain has converged. This is the idea of *contrastive divergence algorithm*, which we briefly discuss in Section 14.4.4.

14.4 Other Applications

In this section, we will review additional applications of stochastic approximation and SGD, giving a preference to breadth over depth.

14.4.1 Online EM algorithm

The EM algorithm (Dempster et al., 1977) is a numerically stable procedure to compute the maximum-likelihood estimator in latent variable models. Slightly changing the notation of previous sections, let X_n denote the latent variable, let Y_n denote the outcome distributed conditional on X_n, and assume model parameters $\theta_\star$. Also, let $f_{\text{com}}(X_n, Y_n; \theta)$ and $f_{\text{obs}}(Y_n; \theta)$ denote, respectively, the complete data and observed data densities; similarly, ℓ_{com} and ℓ_{obs} denote the respective log-likelihoods. For simplicity, we will assume that f_{com} is an exponential family model in the natural parameterization, that is,

$$f_{\text{com}}(X_n, Y_n; \theta) = \exp\left\{S(X_n, Y_n)^\intercal\theta_\star - A(\theta_\star) + B(X_n, Y_n)\right\} \tag{14.35}$$

for appropriate functions S, A, and B. The Fisher information matrix of complete data for parameter value θ is denoted by $\mathcal{I}_{\text{com}}(\theta) = -\mathbb{E}\left(\nabla\nabla\ell_{\text{com}}(\theta; X_n, Y_n)\right)$; similarly, the Fisher information matrix of observed data is denoted by $\mathcal{I}_{\text{obs}}(\theta) = -\mathbb{E}\left(\nabla\nabla\ell_{\text{obs}}(\theta; Y_n)\right)$. Furthermore, we assume a finite data set, where $\mathbf{Y} = (Y_1, \ldots, Y_N)$ denotes all observed data, and $\mathbf{X} = (X_1, \ldots, X_N)$ denotes all missing data.

The classical EM algorithm proceeds by iterating the following steps:

$$Q(\theta, \theta_{n-1}; \mathbf{Y}) = \mathbb{E}\left(\ell_{\text{com}}(\theta; \mathbf{X}, \mathbf{Y})|\,\theta_{n-1}, \mathbf{Y}\right) \qquad \textbf{E-step} \tag{14.36}$$

$$\theta_n = \arg\max_\theta Q(\theta, \theta_{n-1}; \mathbf{Y}) \qquad \textbf{M-step} \tag{14.37}$$

Dempster et al. (1977) showed that the EM algorithm converges to the maximum likelihood estimator $\hat{\theta} = \arg\max_\theta \ell_{\text{obs}}(\theta; \mathbf{Y})$, and that EM is an ascent algorithm, that is, the likelihood is strictly increasing at each iteration. Despite this highly desirable numerical stability, the EM algorithm is impractical for estimation with large data sets because it involves expensive operations, both in the expectation and maximization steps, that need to performed on the entire set of N data points.

To speed up the EM algorithm, Titterington (1984) considered a procedure defined through the iteration

$$\theta_n = \theta_{n-1} + a_n \mathcal{I}_{\text{com}}(\theta_{n-1})^{-1} \nabla \ell_{\text{obs}}(\theta_{n-1}; Y_n) \tag{14.38}$$

This procedure is essentially Sakrison's recursive estimation method described in Section 14.2.1, appropriately modified to use the Fisher information matrix of observed data. In the univariate case, Titterington (1984) applied Fabian's theorem (Fabian, 1968) to show that the estimate in Equation 14.38 satisfies $\sqrt{n}(\theta_n - \theta_\star) \sim \mathcal{N}(0, \mathcal{I}_{\text{com}}(\theta_\star)^{-2} \mathcal{I}_{\text{obs}}(\theta_\star) / (2\mathcal{I}_{\text{obs}}(\theta_\star)\mathcal{I}_{\text{com}}(\theta_\star)^{-1} - 1)$. Thus, as in the classical EM algorithm, the efficiency of Titterington's method (Equation 14.38) depends on the fraction of missing information. Notably, Lange (1995) considered single Newton–Raphson steps in the M-step of the EM algorithm, and derived a procedure that is similar to Equation 14.38.

However, the procedure 14.38 is essentially an explicit stochastic gradient method, and, unlike EM, it can have serious stability and convergence problems. In the exponential family model (Equation 14.35), Nowlan (1991) considered the first true "online" EM algorithm as follows:

$$\begin{aligned} S_n &= (1 - \alpha) S_{n-1} + \alpha \mathbb{E}\left(S(X_n, Y_n) \middle| \theta_{n-1}, Y_n\right) && \textbf{E-step} \\ \theta_n &= \arg\max_\theta \ell_{\text{com}}(\theta; S_n) && \textbf{M-step} \end{aligned} \tag{14.39}$$

where $\alpha \in (0, 1)$. In words, algorithm 14.39 starts from some initial sufficient statistic S_0 and then uses stochastic approximation with a constant step-size α to update it. The maximization step is identical to that of classical EM, and it is more stable than procedure 14.38 because, as iterations proceed, S_n accumulates information over the entire data set. A variant of Nowlan's method with a decreasing step-size was later developed by Sato and Ishii (2000) as follows:

$$\begin{aligned} S_n &= (1 - a_n) S_{n-1} + a_n \mathbb{E}\left(S(X_n, Y_n) \middle| \theta_{n-1}, Y_n\right) && \textbf{E-step} \\ \theta_n &= \arg\max_\theta \ell_{\text{com}}(\theta; S_n) && \textbf{M-step} \end{aligned} \tag{14.40}$$

By the theory of stochastic approximation, procedure 14.40 converges to the observed data maximum-likelihood estimate $\hat{\theta}$. In contrast, procedure 14.39 will not converge with a constant α; it will rather reach a point in the vicinity of $\hat{\theta}$ more rapidly than Equation 14.40, and then oscillate around $\hat{\theta}$. Further online EM algorithms have been developed by several authors (Neal and Hinton, 1998, Cappé and Moulines, 2009). Examples of a growing body of applications of such methods can be found in works by Neal and Hinton (1998), Sato and Ishii (2000), Liu et al. (2006), and Cappé (2011).

14.4.2 MCMC sampling

As before, we need to slightly extend our notation to a Bayesian setting. Let θ denote model parameters with an assumed prior distribution $\pi(\theta)$. A common task in Bayesian inference is to sample from the posterior distribution $f(\theta|\mathbf{Y}) \propto \pi(\theta) f(\mathbf{Y}|\theta)$, given N observed data points $\mathbf{Y} = \{Y_1, \ldots, Y_N\}$.

The Hamiltonian Monte Carlo (HMC) (Neal, 2011) is an MCMC method in which auxiliary parameters p are introduced to improve sampling from $f(\theta|\mathbf{Y})$. In the augmented parameter space, we consider a function $H(\theta, p) = U(\theta) + K(p) \in \mathbb{R}^+$, where $U(\theta) = -\log f(\theta|\mathbf{Y})$ and $K(p) = (1/2)p^\intercal Mp$, M being positive definite. Next, we consider the density

$$h(\theta, p|\mathbf{Y}) = \exp\{-H(\theta, p)\} = \exp\{-U(\theta) - K(p)\} = f(\theta|\mathbf{Y}) \times \mathcal{N}(p, M^{-1})$$

In this parameterization, the variables p are independent of θ. Assuming an initial state (θ_0, p_0), sampling with HMC proceeds in iterations indexed by $n = 1, \ldots$, as follows:

1. Sample $p^* \sim \mathcal{N}(0, M^{-1})$.
2. Using *Hamiltonian dynamics*, compute $(\theta_n, p_n) = \text{ODE}(\theta_{n-1}, p^*)$.
3. Perform a Metropolis–Hastings step for the proposed transition $(\theta_{n-1}, p^*) \to (\theta_n, p_n)$ with acceptance probability $\min[1, \exp(-H(\theta_n, p_n) + H(\theta_{n-1}, p^*)]$.

Step 2 is the key idea in HMC. The parameters (θ, p) are mapped to a physical system, where θ is the position of the system, and p is the momentum. The potential of the physical system is $U(\theta)$ and its kinetic energy is $K(p)$. Function H is known as the *Hamiltonian*. The Hamiltonian dynamics refer to a set of ordinary differential equations (ODE) that govern the movement of the system, and thus determine the future values of (θ, p) given a pair of current values. Being a closed physical system, the Hamiltonian of the system, $H(\theta, p) = U(\theta)+K(p)$, is constant. Thus, in Step 3 of HMC it holds that $-H(\theta_n, p_n)+H(\theta_{n-1}, p^*) = 0$, and thus the acceptance probability is 1, assuming that the solution of the ODE is exact. This is a significant improvement over generic Metropolis–Hastings, where it is usually hard to achieve high acceptance probabilities.

A special case of HMC, known as *Langevin dynamics* (Girolami and Calderhead, 2011), defines the sampling iterations as follows:

$$\begin{aligned} &\eta_n \sim \mathcal{N}(0, \epsilon I) \\ &\theta_n = \theta_{n-1} + \frac{\epsilon}{2}\left(\nabla \log \pi(\theta_{n-1}) + \nabla \log f(\mathbf{Y}|\theta_{n-1})\right) + \eta_n \end{aligned} \tag{14.41}$$

The sampling procedure 14.41 follows from HMC by a numerical solution of the ODE in Step 2 of the algorithm using the *leapfrog* method (Neal, 2011). Parameter $\epsilon > 0$ in Equation 14.41 determines the size of the leapfrog in the numerical solution of Hamiltonian differential equations.

Welling and Teh (2011) studied a simple modification of Langevin dynamics (Equation 14.41) using a stochastic gradient as follows:

$$\begin{aligned} &\eta_n \sim \mathcal{N}(0, \epsilon_n) \\ &\theta_n = \theta_{n-1} + \frac{\epsilon_n}{2}(\nabla \log \pi(\theta_{n-1}) + (N/b) \sum_{i \in \text{batch}} \nabla \log f(Y_i|\theta_{n-1})) + \eta_n \end{aligned} \tag{14.42}$$

The step sizes $\epsilon_n > 0$ satisfy the typical Robbins–Monro requirements, that is, $\sum \epsilon_i = \infty$ and $\sum \epsilon_i^2 < \infty$. Procedure 14.42 is using stochastic gradients averaged over a *batch* of b data points, a technique usually employed in SGD to reduce noise in stochastic gradients. Sato and Nakagawa (2014) proved that procedure 14.42 converges to the true posterior $f(\theta|\mathbf{Y})$ using an elegant theory of stochastic calculus. Sampling through stochastic gradient Langevin dynamics has since generated a lot of related work in posterior sampling for large data sets, and it is still a rapidly expanding research area with contributions from various disciplines (Hoffman et al., 2013, Korattikara et al., 2014, Pillai and Smith, 2014).

14.4.3 Reinforcement learning

Reinforcement learning is the multidisciplinary study of how autonomous agents perceive, learn, and interact with their environment (Bertsekas and Tsitsiklis, 1995). Typically, it is assumed that time t proceeds in discrete steps, and at every step an *agent* is at state $x_t \in \mathcal{X}$, where $\mathcal{X}$ is the state space. On entering a state x_t, two things happen. First, an agent receives a probabilistic *reward* $R(x_t) \in \mathbb{R}$, and, second, the agent takes an *action* $a \in \mathcal{A}$, where $\mathcal{A}$ denotes the action space. This action is determined by the agent's *policy*, which is a function $\pi : \mathcal{X} \to \mathcal{A}$, mapping a state to an action. Nature then decides a *transition* to state x_{t+1} according to a probability that is unknown to the agent.

One important task in reinforcement learning is to estimate the *value function* $V^\pi(x)$, which quantifies the expected value of a specific state $x \in \mathcal{X}$ with respect to policy π, defined as

$$V^\pi(x) = \mathbb{E}\left(R(x)\right) + \gamma\mathbb{E}\left(R(x_1)\right) + \gamma^2\mathbb{E}\left(R(x_2)\right) + \cdots \tag{14.43}$$

where

x_t denotes the state that will be reached starting at x after t transitions

$\gamma \in (0, 1)$ is a parameter that discounts future rewards

Uncertainty in $R(x_t)$ includes the uncertainty of the state x_t because of the stochasticity in state transitions, and the uncertainty from the reward distribution. Thus, $V^\pi(x)$ admits a recursive definition as follows:

$$V^\pi(x) = \mathbb{E}\left(R(x)\right) + \gamma\mathbb{E}\left(V^\pi(x_1)\right) \tag{14.44}$$

When the state is a high-dimensional vector, one popular approach is to use a linear approximation for $V(x)$, such that $V(x) = \theta_\star^\intercal\phi(x)$, where $\phi(x)$ maps a state to a *feature space* with fewer dimensions, and $\theta_\star$ is a vector of fixed parameters. If the agent is at state x_t, then the recursive equation (Equation 14.44) can be rewritten as

$$\mathbb{E}\left(R(x_t) - (\theta_\star^\intercal\phi_t - \gamma\theta_\star^\intercal\phi_{t+1})|\,\phi_t\right) = 0 \tag{14.45}$$

where we set $\phi_t = \phi(x_t)$ for notational convenience. Similar to SGD, this suggests a stochastic approximation method to estimate $\theta_\star$ through the following iteration:

$$\theta_{t+1} = \theta_t + a_t\left[R(x_t) - (\theta_t^\intercal\phi_t - \gamma\theta_t^\intercal\phi_{t+1})\right]\phi_t \tag{14.46}$$

where a_t is a learning rate sequence that satisfies the Robbins–Monro conditions of Section 14.2. Procedure 14.46 is known as the *temporal differences* (TD) learning algorithm (Sutton, 1988). Implicit versions of this algorithm have recently emerged to solve the known stability issues of the classical TD algorithm (Wang and Bertsekas, 2013, Tamar et al., 2014). For example, Tamar et al. (2014) consider computing the term $\theta_t^\intercal\phi_t$ at the future iterate, the resulting *implicit* TD algorithm being defined as

$$\theta_{t+1} = (I + a_t\phi_t\phi_t^\intercal)^{-1}\left[\theta_t + a_t(R(x_t) + \gamma\theta_t^\intercal\phi_{t+1})\phi_t\right] \tag{14.47}$$

Similar to implicit SGD, iteration 14.47 stabilizes the TD iteration 14.46. With the advent of online multiagent markets, methods and applications in reinforcement learning have been receiving a renewed stream of research effort (Gosavi, 2009).

14.4.4 Deep learning

Deep learning is the task of estimating parameters of statistical models that can be represented by multiple layers of nonlinear operations, such as neural networks (Bengio,

2009). Such models, also referred to as *deep architectures*, consist of *units* that can perform a basic prediction task, and are grouped in layers such that the output of one layer forms the input of another layer that sits directly on top. Furthermore, the models are usually augmented with *latent units* that are defined to represent structured quantities of interest, such as edges or shapes in an image.

One basic building block of deep architectures is the Restricted Boltzmann Machine (RBM). The complete-data density for one data point (X, Y) of the states of hidden and observed input units, respectively, is given by

$$f(X, Y; \theta) = \frac{\exp\{-b'Y - c'x - X'WY\}}{Z(\theta)} \tag{14.48}$$

where $\theta = (b, c, W)$ are the model parameters, and the function $Z(\theta) = \sum_{X,Y} \exp\{-b'Y - c'x - X'WY\}$, also known as the partition function, acts as the normalizing constant. Furthermore, the sample spaces for X and Y are discrete (for example, binary) and finite. The observed-data density is thus $f(Y; \theta) = \sum_X f(X, Y; \theta)$. Let $H(X, Y; \theta) = b'Y + c'x + X'WY$, such that $f(X, Y; \theta) = (e^{-H(X,Y;\theta)})/(Z(\theta))$. Consider also observed data $\mathbf{Y} = \{Y_1, Y_2, \ldots, Y_N\}$, and missing data $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$.

Through simple algebra one can obtain the gradient of the log-likelihood of observed data in the following convenient form:

$$\nabla \ell(\theta; \mathbf{Y}) = -\left[\mathbb{E}\left(\nabla H(\mathbf{X}, \mathbf{Y}; \theta)\right) - \mathbb{E}\left(\nabla H(\mathbf{X}, \mathbf{Y}; \theta) | \mathbf{Y}\right)\right] \tag{14.49}$$

where $H(\mathbf{X}, \mathbf{Y}; \theta) = \sum_{n=1}^{N} H(X_n, Y_n; \theta)$. In practical situations, the data points (X_n, Y_n) are binary. Therefore, the conditional distribution of the missing data $X_n|Y_n$ is readily available through a logistic regression model, and thus the second term of Equation 14.49 is easy to sample from. Similarly, $Y_n|X_n$ is easy to sample from. However, the first term in Equation 14.49 requires sampling from the joint distribution of the complete data $(\mathbf{X}, \mathbf{Y})$, which conceptually is easy to do using the aforementioned conditionals and a Gibbs sampling scheme (Geman and Geman, 1984). However, the domain for both $\mathbf{X}$ and $\mathbf{Y}$ is typically very large, for example, it comprises thousands or millions of units, and thus a full Gibbs on the joint distribution is impossible.

The method of *contrastive divergence* (Hinton, 2002, Carreira-Perpinan and Hinton, 2005) has been applied for training such models with considerable success. The algorithm proceeds as follows for steps $i = 1, 2, \ldots$:

1. Sample one state $Y^{(i)}$ from the empirical distribution of observed data $\mathbf{Y}$.
2. Sample $X^{(i)}|Y^{(i)}$, that is, the hidden state.
3. Sample $Y^{(i,\text{new})}|X^{(i)}$.
4. Sample $X^{(i,\text{new})}|Y^{(i,\text{new})}$.
5. Evaluate the gradient (Equation 14.49) using $(X^{(i)}, Y^{(i)})$ for the second term, and the sample $(X^{(i,\text{new})}, Y^{(i,\text{new})})$ for the first term.
6. Update the parameters in θ using constant-step-size SGD and the estimated gradient from Step 5.

In other words, contrastive divergence attempts to estimate $\nabla \ell(\theta; \mathbf{Y})$ in Equation 14.49. This estimation is biased because $(X^{(i,\text{new})}, Y^{(i,\text{new})})$ is assumed to be from the exact joint distribution of (X, Y); however, they are single Gibbs iterations starting from the observed and imputed data $(X^{(i)}, Y^{(i)})$, respectively. In theory, Steps 3 and 4 could be repeated k times; for example, if $k \to \infty$ the sampling distribution of $(X^{(i,\text{new})}, Y^{(i,\text{new})})$ would be the exact joint distribution of (X, Y), leading to unbiased estimation of $\nabla \ell(\theta; \mathbf{Y})$ of

Equation 14.49. Surprisingly, it has been empirically observed that $k = 1$ is enough for good performance in many learning tasks (Hinton, 2002, Taylor et al., 2006, Salakhutdinov et al., 2007, Bengio, 2009, Bengio and Delalleau, 2009), which is a testament to the power and flexibility of stochastic gradient methods.

14.5 Glossary

SGD: Stochastic gradient descent.

References

Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276.

Amari, S.-I., Chen, T.-P., and Cichocki, A. (1997). Stability analysis of learning algorithms for blind source separation. *Neural Networks*, 10(8):1345–1351.

Amari, S.-I., Park, H., and Fukumizu, K. (2000). Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation*, 12(6):1399–1409.

Bach, F. and Moulines, E. (2013). Non-strongly-convex smooth stochastic approximation with convergence rate $o(1/n)$. In *Advances in Neural Information Processing Systems*, pp. 773–781.

Bather, J. (1989). *Stochastic Approximation: A Generalisation of the Robbins-Monro Procedure*, volume 89. Mathematical Sciences Institute, Cornell University, Ithaca, NY.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127.

Bengio, Y. and Delalleau, O. (2009). Justifying and generalizing contrastive divergence. *Neural Computation*, 21(6):1601–1621.

Bertsekas, D. P. (2011). Incremental proximal methods for large scale convex optimization. *Mathematical Programming*, 129(2):163–195.

Bertsekas, D. P. and Tsitsiklis, J. N. (1995). Neuro-dynamic programming: An overview. In *Proceedings of the 34th IEEE Conference on Decision and Control*, volume 1, pp. 560–564. IEEE, New Orleans, LA.

Blum, J. R. (1954). Multidimensional stochastic approximation methods. *Annals of Mathematical Statistics*, 25:737–744.

Bordes, A., Bottou, L., and Gallinari, P. (2009). SGD-QN: Careful quasi-Newton stochastic gradient descent. *The Journal of Machine Learning Research*, 10:1737–1754.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, Berlin, Germany.

Bottou, L. and Le Cun, Y. (2005). Online learning for very large data sets. *Applied Stochastic Models in Business and Industry*, 21(2):137–151.

Bottou, L. and Murata, N. (2002). Stochastic approximations and efficient learning. *The Handbook of Brain Theory and Neural Networks*, 2nd edition. The MIT Press, Cambridge, MA.

AQ 1

Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19:577–593.

Cappé, O. (2011). Online EM algorithm for Hidden Markov models. *Journal of Computational and Graphical Statistics*, 20(3):728–749.

Cappé, O. and Moulines, E. (2009). Online Expectation–Maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B* (*Statistical Methodology*), 71(3):593–613.

Carreira-Perpinan, M. A. and Hinton, G. E. (2005). On contrastive divergence learning. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pp. 33–40. CiteSeer.

AQ 2

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

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38.

Duchi, J., Hazan, E., and Singer, Y. (2011a). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Duchi, J., Hazan, E., and Singer, Y. (2011b). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 999999:2121–2159.

AQ 3

Dupuis, P. and Simha, R. (1991). On sampling controlled stochastic approximation. *IEEE Transactions on Automatic Control*, 36(8):915–924.

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

Fabian, V. (1973). Asymptotically efficient stochastic approximation; the RM case. *Annals of Statistics*, 1:486–495.

Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222:309–368.

Fisher, R. A. (1925). *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh, Scotland.

Gardner, W. A. (1984). Learning characteristics of stochastic gradient descent algorithms: A general study, analysis, and critique. *Signal Processing*, 6(2):113–133.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741.

George, A. P. and Powell, W. B. (2006). Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine Learning*, 65(1):167–198.

Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B* (*Statistical Methodology*), 73(2):123–214.

Gosavi, A. (2009). Reinforcement learning: A tutorial survey and recent advances. *INFORMS Journal on Computing*, 21(2):178–192.

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society. Series B* (*Methodological*), 46:149–192.

Hastie, T., Tibshirani, R., and Friedman, J. (2011). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edition. Springer, New York.

Hennig, P. and Kiefel, M. (2013). Quasi-newton methods: A new direction. *The Journal of Machine Learning Research*, 14(1):843–865.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323.

Karoui, N. E. (2008). Spectrum estimation for large dimensional covariance matrices using random matrix theory. *Annals of Statistics*, 36(6):2757–2790.

Kivinen, J., Warmuth, M. K., and Hassibi, B. (2006). The p-norm generalization of the LMS algorithm for adaptive filtering. *IEEE Transactions on Signal Processing*, 54(5): 1782–1793.

Korattikara, A., Chen, Y., and Welling, M. (2014). Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *Proceedings of The 31st International Conference on Machine Learning*, pp. 181–189.

Krakowski, K. A., Mahony, R. E., Williamson, R. C., and Warmuth, M. K. (2007). A geometric view of non-linear online stochastic gradient descent. *Author Website*. AQ 4

Kulis, B. and Bartlett, P. L. (2010). Implicit online learning. In *Proceedings of the 27th International Conference on Machine Learning* (*ICML-10*), pp. 575–582.

Lai, T. L. and Robbins, H. (1979). Adaptive design and stochastic approximation. *Annals of Statistics*, 7:1196–1221.

Lange, K. (1995). A gradient algorithm locally equivalent to the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57:425–437.

Le Cun, L. B. Y. and Bottou, L. (2004). Large scale online learning. *Advances in Neural Information Processing Systems*, 16:217.

Lehmann, E. L. and Casella, G. (1998). *Theory of Point Estimation*, volume 31. Springer, New York.

Lehmann, E. H. and Casella, G. (2003). *Theory of Point Estimation*, 2nd edition. Springer, New York.

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

Liu, Z., Almhana, J., Choulakian, V., and McGorman, R. (2006). Online EM algorithm for mixture with application to internet traffic modeling. *Computational Statistics & Data Analysis*, 50(4):1052–1071.

Ljung, L., Pflug, G., and Walk, H. (1992). *Stochastic Approximation and Optimization of Random Systems*, volume 17. Springer Basel AG, Basel, Switzerland.

Moulines, E. and Bach, F. R. (2011). Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pp. 451–459.

Murata, N. (1998). A statistical study of online learning. *Online Learning and Neural Networks*. Cambridge University Press, Cambridge.

Nagumo, J.-I. and Noda, A. (1967). A learning method for system identification. *IEEE Transactions on Automatic Control*, 12(3):282–287.

National Research Council (2013). *Frontiers in Massive Data Analysis*. National Academies Press, Washington, DC.

Neal, R. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, volume 2.

Neal, R. M. and Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pp. 355–368. Springer.

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

Nevelson, M. B. and Khasminskiĭ, R. Z. (1973). *Stochastic Approximation and Recursive Estimation*, volume 47. American Mathematical Society, Providence, RI.

Nowlan, S. J. (1991). Soft competitive adaptation: Neural network learning algorithms based on fitting statistical mixtures.

Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231.

AQ 5 Pillai, N. S. and Smith, A. (2014). Ergodicity of approximate MCMC chains with applications to large data sets. arXiv preprint: arXiv:1405.0182.

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

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

Rockafellar, R. T. (1976). Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898.

Rosasco, L., Villa, S., and Vũ, B. C. (2014). Convergence of stochastic proximal gradient algorithm. arXiv preprint: arXiv:1403.5074.

Ruppert, D. (1988a). Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering, Ithaca, NY.

Ruppert, D. (1988b). Stochastic approximation. Technical report, Cornell University Operations Research and Industrial Engineering, Ithaca, NY.

Ryu, E. K. and Boyd, S. (2014). Stochastic proximal iteration: A non-asymptotic improvement upon stochastic gradient descent. *Author website, early draft.*

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

Sakrison, D. J. (1965). Efficient recursive estimation; application to estimating the parameters of a covariance function. *International Journal of Engineering Science*, 3(4):461–483.

Salakhutdinov, R., Mnih, A., and Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pp. 791–798. ACM, New York.

Sato, I. and Nakagawa, H. (2014). Approximation analysis of stochastic gradient Langevin Dynamics by using Fokker-Planck equation and Ito process. *JMLR W&CP*, 32(1): 982–990.

Sato, M.-A. and Ishii, S. (2000). Online EM algorithm for the normalized Gaussian network. *Neural Computation*, 12(2):407–432.

Schaul, T., Zhang, S., and LeCun, Y. (2012). No more pesky learning rates. arXiv preprint: arXiv:1206.1106.

Schmidt, M., Le Roux, N., and Bach, F. (2013). Minimizing finite sums with the stochastic average gradient. Technical report, HAL 00860051.

Schraudolph, N., Yu, J., and Günter, S. (2007). A stochastic quasi-Newton method for online convex optimization.

Slock, D. T. (1993). On the convergence behavior of the LMS and the normalized LMS algorithms. *IEEE Transactions on Signal Processing*, 41(9):2811–2825.

Spall, J. C. (2000). Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transactions on Automatic Control*, 45(10):1839–1853.

Spall, J. C. (2005). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, volume 65. John Wiley & Sons, Hoboken, NJ.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44.

Tamar, A., Toulis, P., Mannor, S., and Airoldi, E. (2014). Implicit temporal differences. In *Neural Information Processing Systems, Workshop on Large-Scale Reinforcement Learning.*

Taylor, G. W., Hinton, G. E., and Roweis, S. T. (2006). Modeling human motion using binary latent variables. In *Advances in Neural Information Processing Systems*, pp. 1345–1352.

Titterington, D. M. (1984). Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46:257–267.

Toulis, P. and Airoldi, E. M. (2014). Implicit stochastic gradient descent for principled estimation with large data sets. arXiv manuscript no. 1408.2923.

Toulis, P. and Airoldi, E. M. (2015). Scalable estimation strategies based on stochastic approximations: Classical results and new insights. *Statistics and Computing*, 25: 781–795.

Toulis, P., Rennie, J., and Airoldi, E. (2014). Statistical analysis of stochastic gradient methods for generalized linear models. *JMLR W&CP*, 32(1):667–675.

Toulis, P., Tran, D., and Airoldi, E. (2015). Stability and optimality in stochastic gradient descent. arXiv: 1505.02417.

Tran, D., Lan, T., Toulis, P., and Airoldi, E. (2015a). *Stochastic Gradient Descent for Scalable Estimation*. R package version 0.1.

Tran, D., Toulis, P., and Airoldi, E. (2015b). The sgd R package for principled estimation with stochastic gradient methods. Manuscript.

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

Wang, C., Chen, X., Smola, A., and Xing, E. (2013). Variance reduction for stochastic gradient optimization. In *Advances in Neural Information Processing Systems*, pp. 181–189.

Wang, M. and Bertsekas, D. P. (2013). Stabilization of stochastic iterative methods for singular and nearly singular linear systems. *Mathematics of Operations Research*, 39(1): 1–30.

Wei, C. (1987). Multivariate adaptive stochastic approximation. *Annals of Statistics*, 15:1115–1130.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 681–688.

Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. *IRE WESCON Convention Record*, 4:96–104. (Defense Technical Information Center.)

Xiao, L. and Zhang, T. (2014). A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24:2057–2075.

Xu, W. (2011). Towards optimal one pass large scale learning with averaged stochastic gradient descent. arXiv preprint: arXiv:1107.2490.

Younes, L. (1999). On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics*, 65(3–4):177–228.

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-First International Conference on Machine Learning*, p. 116. ACM, New York.

Author Query Sheet

Chapter No: C014

Query No.	Queries	Response
AQ 1	Please provide editor names for "Bertsekas and Tsitsiklis (1995); Bottou (2010); Carreira-Perpinan and Hinton (2005); Johnson and Zhang (2013); Korattikara et al. (2014); Kulis and Bartlett (2010); Lange (1995); Le Cun and Bottou (2004); Neal and Hinton (1998); Salakhutdinov et al. (2007); Taylor et al. (2006); Wang et al. (2013); Welling and Teh (2011). Zhang (2004)."	
AQ 2	Please provide publisher name or publisher location or both, for "Carreira-Perpinan and Hinton (2005); Neal (2011); Neal and Hinton (1998); Schmidt et al. (2013); Tamar et al. (2014); Welling and Teh (2011); Moulines and Bach (2011)."	
AQ 3	Duchi et al. (2011a) and Duchi et al. (2011b) seem to be the same except volume number (999999). Please check and confirm if "Duchi et al. (2011b)" can be deleted.	
AQ 4	Please provide the complete details of "Krakowski et al. (2007); Nowlan (1991); Ryu and Boyd (2014); Schraudolph et al. (2007); Tran et al. (2015a); Tran et al. (2015b)."	
AQ 5	Please provide the complete details of "Pillai and Smith (2014); Rosasco et al. (2014); Schaul et al. (2012); Toulis and Airoldi (2014); Toulis et al. (2015); Xu (2011)," if already published.	