Published in final edited form as:
*Stat Comput*. 2015 July 1; 25(4): 781–795. doi:10.1007/s11222-015-9560-y.

# Scalable estimation strategies based on stochastic approximations: Classical results and new insights

**Edoardo M. Airoldi** and
Harvard University, Department of Statistics

**Panos Toulis**
Harvard University, Department of Statistics

Edoardo M. Airoldi: airoldi@fas.harvard.edu; Panos Toulis: ptoulis@fas.harvard.edu

## Abstract

Estimation with large amounts of data can be facilitated by stochastic gradient methods, in which model parameters are updated sequentially using small batches of data at each step. Here, we review early work and modern results that illustrate the statistical properties of these methods, including convergence rates, stability, and asymptotic bias and variance. We then overview modern applications where these methods are useful, ranging from an online version of the EM algorithm to deep learning. In light of these results, we argue that stochastic gradient methods are poised to become benchmark principled estimation procedures for large data sets, especially those in the family of stable proximal methods, such as implicit stochastic gradient descent.

### Keywords

maximum likelihood; exponential family; stochastic gradient descent methods; implicit stochastic gradient descent; recursive estimation; efficient estimation; optimal learning rate; asymptotic analysis; big data

## 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 EM algorithm and iteratively reweighted least squares [Fisher, 1925a, Dempster et al., 1977, Green, 1984], are not scalable to modern data sets with hundreds of millions of data points and hundreds of thousands of covariates [National Research Council, 2013].

To illustrate, let us consider the problem of estimating the true vector of parameters $\theta_\star \in \mathbb{R}^p$ from an i.i.d. sample $\boldsymbol{Y} = \{\boldsymbol{y}_n\}$, for $n = 1, 2\ldots N$, where a data point $\boldsymbol{y}_n \in \mathbb{R}^d$ is distributed according to a density $f(\boldsymbol{y}_n; \theta_\star)$ with log-likelihood function $\ell(\boldsymbol{\theta}; \boldsymbol{Y}) = \sum_{n=1}^{N} \log f(\boldsymbol{y}_n; \boldsymbol{\theta})$. Traditional estimation methods are typically *iterative* and have a running-time complexity that ranges between $\mathcal{O}(Np^3)$ and $\mathcal{O}(Np)$, in worst cases and best cases respectively. Newton-Raphson methods, for instance, update an estimate $\boldsymbol{\theta}_{n-1}^{\text{nr}}$ of the parameters through the recursion

$$\boldsymbol{\theta}_n^{\text{nr}}=\boldsymbol{\theta}_{n-1}^{\text{nr}}-\boldsymbol{H}_{n-1}^{-1}\nabla\ell(\boldsymbol{\theta}_{n-1}^{\text{nr}};\boldsymbol{Y}), \quad (1)$$

where $\boldsymbol{H}_n=\nabla\nabla\ell(\boldsymbol{\theta}_n^{\text{nr}};\boldsymbol{Y})$ is the $p \times p$ Hessian matrix of the log-likelihood. The matrix inversion and the likelihood computation yield an algorithm with roughly $\mathcal{O}(Np^{2+\varepsilon})$ complexity which makes it unsuitable for large data sets. Fisher scoring replaces the Hessian matrix with its expected value i.e., it uses the Fisher information matrix $\mathcal{I}(\theta) = -\mathbb{E}(\nabla\nabla\ell(\theta; \boldsymbol{y}_n))$, where the expectation is over the random sample $\boldsymbol{y}_n$. The advantage of this method is that a steady increase in the likelihood is possible, as in the EM algorithm, since $\mathcal{I}(\theta)$ is positive-definite, and thus the difference

$$\ell(\boldsymbol{\theta}+\varepsilon\Delta\boldsymbol{\theta};\boldsymbol{Y})-\ell(\boldsymbol{\theta};\boldsymbol{Y}) \approx \varepsilon\,\ell(\boldsymbol{\theta};\boldsymbol{Y})^\top \mathscr{I}(\boldsymbol{\theta})^{-1}\ell(\boldsymbol{\theta};\boldsymbol{Y})+\mathscr{O}(\varepsilon^2)$$

can be made positive for an appropriately small value $\varepsilon > 0$. However, Fisher scoring performs very similarly to Newton-Raphson in practice, and the two algorithms are actually identical in the exponential family [Lange, 2010]. Furthermore, Fisher scoring is computationally comparable to Newton-Raphson and thus unsuited for problems with large data sets.

Quasi-Newton (QN) methods are a powerful alternative and are widely used in practice. In QN methods, the Hessian is approximated by a low-rank matrix that is updated at each iteration as new values of the gradient become available, thus yielding algorithms with complexity $\mathcal{O}(Np^2)$ or $\mathcal{O}(Np)$ in certain favorable cases [Hennig and Kiefel, 2013]. Other general estimation algorithms such as EM or iteratively reweighted least squares [Green, 1984] involve computations (e.g. inversions or maximizations between iterations) that are significantly more expensive than QN methods.

However, estimation with massive data sets requires a running time complexity that is roughly $\mathcal{O}(Np^{1-\varepsilon})$ i.e., that is linear in $N$ but sublinear in the parameter dimension $p$. The first requirement on $N$ seems hard to overcome since an iteration over all data points needs to be performed, at least when data are i.i.d.; thus, sublinearity in $p$ is crucial [Bousquet and Bottou, 2008]. Such computational requirements have recently sparked interest in algorithms that utilize only *first-order* information i.e., methods that utilize only gradient computations.[1] Such performance is achieved by the *stochastic gradient descent* (SGD) algorithm, which was initially proposed by Sakrison [1965] as a *recursive estimation method*, albeit not in first-order form. A typical first-order SGD is defined by the iteration

[1] Second-order methods typically use the Hessian matrix of second-order derivatives of the log-likelihood and are discussed in detail in Section 3.

$$\boldsymbol{\theta}_n^{\mathrm{sgd}} = \boldsymbol{\theta}_{n-1}^{\mathrm{sgd}} + a_n \nabla \ell(\boldsymbol{\theta}_{n-1}^{\mathrm{sgd}}; \boldsymbol{y}_n). \quad (2)$$

We will refer to Equation (2) as SGD with *explicit updates*, or *explicit SGD* for short, because the next iterate $\boldsymbol{\theta}_n^{\mathrm{sgd}}$ can be computed immediately after the new data point $\boldsymbol{y}_n$ is observed.[2] The sequence $a_n > 0$ is a carefully chosen *learning rate* sequence which is 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 (2) is appealing because the expensive inversion of $p \times p$ matrices, as in Newton-Raphson, is replaced by a single sequence $a_n > 0$. Furthermore, the log-likelihood is evaluated at a single observation $\boldsymbol{y}_n$, and not on the entire data set $\boldsymbol{Y}$. Necessarily this incurs information loss which is important to quantify. From a theoretical perspective the explicit SGD updates are justified because, under typical regularity conditions, $\mathbb{E}(\nabla \ell(\theta_\star; \boldsymbol{y}_n)) = 0$ and thus $\theta_n \to \theta_\star$ by the properties of the Robbins-Monro procedure [Robbins and Monro, 1951]. However, the explicit SGD procedure requires careful tuning of the learning rate parameter; small values of $\alpha$ will make the iteration (2) very slow to converge, whereas for large values of $\alpha$ explicit SGD will either have a large asymptotic variance, or even diverge numerically. As a recursive estimation method, explicit SGD was first proposed by Sakrison (1965) and has attracted attention in the machine learning community as a fast prediction method for large-scale problems [Le Cun and Bottou, 2004, Zhang, 2004].

In order to stabilize explicit SGD without sacrificing computational efficiency, Toulis et al. [2014] defined the *implicit SGD* procedure through the iteration

$$\boldsymbol{\theta}_n^{\mathrm{im}} = \boldsymbol{\theta}_{n-1}^{\mathrm{im}} + a_n \nabla \ell(\boldsymbol{\theta}_n^{\mathrm{im}}; \boldsymbol{y}_n). \quad (3)$$

Note that Equation (3) is *implicit* because the next iterate $\boldsymbol{\theta}_n^{\mathrm{im}}$ appears in both sides of the equation.[3] This simple tweak of the explicit SGD procedure has quite remarkable statistical properties. In particular, assuming a common starting point $\boldsymbol{\theta}_{n-1}^{\mathrm{sgd}} = \boldsymbol{\theta}_{n-1}^{\mathrm{im}} \triangleq \boldsymbol{\theta}$, one can show through a simple Taylor approximation of (3) around $\theta$, that the implicit update satisfies

$$\Delta \boldsymbol{\theta}_n^{\mathrm{im}} = \left(\boldsymbol{I} + a_n \hat{\mathcal{I}}(\boldsymbol{\theta}; \boldsymbol{y}_n)\right)^{-1} \Delta \boldsymbol{\theta}_n^{\mathrm{sgd}} + \mathcal{O}(a_n^2), \quad (4)$$

where $\Delta\theta_n = \theta_n - \theta_{n-1}$ for both methods, and $\hat{\mathcal{I}}(\theta; \boldsymbol{y}_n) = -\nabla\nabla \ell(\theta; \boldsymbol{y}_n)$ is the *observed* Fisher information matrix. Thus, the implicit SGD procedure calculates updates that are a *shrinked* version of the explicit ones. In contrast to explicit SGD, implicit SGD is significantly more stable in small-samples, and it is also robust to misspecifications of the learning rate

[2] Procedure (2) is actually an ascent algorithm because it aims to maximize the log-likelihood, and thus a more appropriate name would be stochastic gradient ascent. However, we will use the term “descent” in order to keep in line with the relevant optimization literature, which traditionally considers minimization problems through descent algorithms.

[3] The solution of the fixed-point equation (3) requires additional computations per iterations. However, Toulis et al. [2014] derive a computationally efficient implicit algorithm in the context of generalized linear models. Furthermore, approximate solutions of implicit updates are possible for any statistical model (see Equation (4)).

parameter $a$. Furthermore, implicit SGD computes iterates that belong in the support of the parameter space, whereas explicit SGD would normally require an additional projection step. 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 (3) and was shown to be consistent and robust to input noise [Slock, 1993]. Theoretical justification for implicit SGD comes either from implicit variations of the Robbins-Monro procedure [Toulis et al., 2014], or through *proximal methods* in optimization [Parikh and Boyd, 2013], such as mirror-descent [Nemirovski, 1983, Beck and Teboulle, 2003]. Assuming differentiability of the log-likelihood, the implicit SGD update (3) can be expressed as a proximal method through the solution of

$$\boldsymbol{\theta}_n^{\mathrm{im}} = \arg\max_{\boldsymbol{\theta}} \left\{ -\frac{1}{2} \left\| \boldsymbol{\theta} - \boldsymbol{\theta}_{n-1}^{\mathrm{im}} \right\|^2 + a_n \ell(\boldsymbol{\theta}; \boldsymbol{y}_n) \right\}, \quad (5)$$

where the right-hand side is the *proximal operator*. The update in Equation (5) is the stochastic version of the deterministic proximal point algorithm by Rockafellar [1976], and has been analyzed recently, in various forms, for convergence and stability [Ryu and Boyd, Rosasco et al., 2014] Recent work has established the consistency of certain implicit methods similar to (3) [Kivinen and Warmuth, 1995, Kivinen et al., 2006, Kulis and Bartlett, 2010] and their robustness has been useful in a range of modern machine learning problems [Nemirovski et al., 2009, Kulis and Bartlett, 2010, Schuurmans and Caelli, 2007].

The structure of this chapter is as follows. In Section 2 we give an overview of the Robbins-Monro procedure and Sakrison's recursive estimation method, which form the theoretical basis of SGD methods; we further provide a quick overview of early results on the statistical efficiency of the aforementioned methods. In Section 3, we formally introduce explicit and implicit SGD, and treat those procedures as *statistical estimation methods* that provide an estimator $\theta_n$ of the model parameters $\theta_\bigstar$ after $n$ iterations. In Section 3.1 we give results on the frequentist statistical properties of SGD estimators i.e., their asymptotic bias and asymptotic variance across multiple realizations of the data set $\boldsymbol{Y}$. We then leverage those results to study optimal learning rate sequences $a_n$ (Section 3.4), the loss of statistical efficiency in SGD and ways to fix it through reparameterization (Section 3.3). We briefly discuss stability in Section 3.2. In Section 3.5, we present significant extensions to first-order SGD, namely averaged SGD, variants of second-order SGD, and Monte-Carlo SGD. Finally, in Section 4, we review significant applications of SGD in various areas of statistics and machine learning, namely in online EM, MCMC posterior sampling, reinforcement learning and deep learning.

# 2 Stochastic approximations

## 2.1 Robbins and Monro's procedure

Consider the one-dimensional setting where one data point is denoted by $y_n \in \mathbb{R}$ and it is controlled by a parameter $\theta$ with regression function $M(\theta) = \mathbb{E}(y| \theta)$ that is non nondecreasing, and whose analytic form might be unknown. Robbins and Monro [1951] considered the problem of finding the unique point $\theta_\bigstar$ for which $M(\theta_\bigstar) = 0$. They devised a procedure, known as the *Robbins-Monro procedure*, in which an estimate $\theta_{n-1}$ of $\theta_\star$ is utilized to sample one new data point $y_n$ such that $\mathbb{E}(y_n | \theta_{n-1}) = M(\theta_{n-1})$; the estimate is then updated according to the following simple rule:

$$\theta_n = \theta_{n-1} - a_n y_n. \quad (6)$$

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

**a.** $(x - \theta_\star)M(x) > 0$ for $x$ in a neighborhood of $\theta_\star$,

**b.** $\mathbb{E}(y_n^2 | \theta) < \infty$ for any $\theta$, and

**c.** $\sum_{i=1}^{\infty} a_i = \infty$ and $\sum_{i=1}^{\infty} a_i^2 < \infty$.

The original proof is technical but the main idea is straightforward. Let $b_n \triangleq \mathbb{E}((\theta_n - \theta_\star)^2)$ denote the squared error, then through iteration (6) one can obtain

$$b_n = b_{n-1} - 2a_n \mathbb{E}((\theta_{n-1} - \theta_\star) M(\theta_{n-1})) + a_n^2 \mathbb{E}(y_n^2). \quad (7)$$

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

$$b_n = (1 - 2a_n M'(\theta_\star)) b_{n-1} + a_n^2 \mathbb{E}(y_n^2). \quad (8)$$

For a learning rate sequence of the form $a_n = \alpha/n$ typical proof techniques in stochastic approximation [Chung, 1954] can establish that $b_n \rightarrow 0$. Furthermore, it holds $nb_n \rightarrow \alpha^2\sigma^2(2\alpha M'(\theta_\star) - 1)^{-1}$ where $\sigma^2 \triangleq \mathbb{E}(y_n^2 | \theta_\star)$ when this limit exists; this result was not given in the original paper by Robbins and Monro [1951] but it was soon derived by several other authors [Chung, 1954, Sacks, 1958, Fabian, 1968a]. Thus, the learning parameter $\alpha$ is critical for the performance of the Robbins-Monro procedure. Its optimal value is $\alpha_\star = 1/M'(\theta_\star)$, which requires knowledge of the slope of $M(\cdot)$ at the true parameter values. In the multidimensional case the efficiency of stochastic approximations -including stochastic gradient descent- depend on the Jacobian of the mean value function of the statistic used in the iterations (see Section 3.1). This early result spawned 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 of the stochastic process (e.g., the quantity $M'(\theta_\star)$) are also being estimated along the way.

### 2.2 Sakrison's recursive estimation method

Although initially applied in sequential experiment design, the Robbins-Monro procedure was soon adapted for estimation. Sakrison [1965] was interested in estimating the parameters $\boldsymbol{\theta}_\star$ of a model that generated i.i.d. observations $\boldsymbol{y}_n$ in a way that is computationally and statistically efficient, similar to our setup in the introduction. He recognized that the statistical identity $\mathbb{E}(\nabla \ell(\boldsymbol{\theta}_\star; \boldsymbol{y}_n)) = 0$, where the expectation is over the observed data $\boldsymbol{y}_n$, provides the theoretical basis for a general estimation method using the

Robbins-Monro procedure. Sakrison's *recursive estimation method* was essentially one of the first *explicit* SGD method proposed in the literature:

$$\boldsymbol{\theta}_n^{\mathrm{sak}} \approx \boldsymbol{\theta}_{n-1}^{\mathrm{sak}} + (1/n)\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_{n-1}^{\mathrm{sak}})^{-1}\nabla\ell(\boldsymbol{\theta}_{n-1}^{\mathrm{sak}};\boldsymbol{y}_n), \tag{9}$$

The SGD procedure (9) is second-order since it is using a matrix to condition the gradient of the log-likelihood. Under typical regularity conditions $\boldsymbol{\theta}_n^{\mathrm{sak}} \rightarrow \boldsymbol{\theta}_\star$, and thus $\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_n^{\mathrm{sak}}) \rightarrow \boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)$. Sakrison [1965] also proved that $n\mathbb{E}\left(\|\boldsymbol{\theta}_n^{\mathrm{sak}} - \boldsymbol{\theta}_\star\|^2\right) \rightarrow \mathrm{trace}(\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)^{-1})$, and so the estimation of $\theta_\bigstar$ is asymptotically efficient under this norm objective. It is interesting to note that updates of the form (9) appeared very early in the statistical literature. For example, Fisher [1925b] suggested that an inefficient estimator $\theta_N$ using $N$ data points can be made asymptotically efficient by considering a new estimator $\boldsymbol{\theta}_N^+ = \boldsymbol{\theta}_N + (1/N)\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)^{-1}\sum_{i=1}^{N}\ell(\boldsymbol{\theta}_N;\boldsymbol{y}_i)$. The surprising result in Sakrison's work was that asymptotically optimal estimation is also possible by using only gradients of the log-likelihood on single data points $\boldsymbol{y}_i$ in the iterated algorithm (9).

## 3 Estimation with stochastic gradient methods

For the rest of this chapter we will consider a simple generalization of explicit and implicit SGD that is similar to Sakrison's method as follows:

$$\boldsymbol{\theta}_n^{\mathrm{sgd}} = \boldsymbol{\theta}_{n-1}^{\mathrm{sgd}} + \boldsymbol{C}_n\nabla\ell(\boldsymbol{\theta}_{n-1}^{\mathrm{sgd}};\boldsymbol{y}_n), \tag{10}$$

$$\boldsymbol{\theta}_n^{\mathrm{im}} = \boldsymbol{\theta}_{n-1}^{\mathrm{im}} + \boldsymbol{C}_n\nabla\ell(\boldsymbol{\theta}_n^{\mathrm{im}};\boldsymbol{y}_n). \tag{11}$$

In general all $\boldsymbol{C}_n$ are symmetric and positive-definite matrices, and serve to stabilize and optimize stochastic iterations as in (10) and (11). In the limit $n\boldsymbol{C}_n \rightarrow \boldsymbol{C}$ where $\boldsymbol{C}$ is a symmetric and positive-definite matrix. If $\boldsymbol{C}_n$ is not trivial (e.g., scaled identity), we will refer to (10) and (11) as *second-order explicit SGD* and *second-order implicit SGD*, respectively. When $\boldsymbol{C}_n = a_n\boldsymbol{I}$ i.e., it is the scaled identity matrix for some sequence $a_n > 0$ satisfying the Robbins-Monro conditions, we will refer to (10) and (11) as *first-order explicit SGD* and *first-order implicit SGD*, respectively; in this case, definitions (10) and (11) are identical to definitions (2) and (3) in the introduction. In some cases, we will consider models in the exponential family under the *natural parameterization* with density

$$f(\boldsymbol{y}_n;\boldsymbol{\theta}_\star) = \exp\{\boldsymbol{\theta}_\star{}^\top\boldsymbol{s}(\boldsymbol{y}_n) - A(\boldsymbol{\theta}_\star) + B(\boldsymbol{y}_n)\}, \tag{12}$$

where $\boldsymbol{s}(\boldsymbol{y}_n)$ is the vector of $p$ sufficient statistics, and $A(\cdot)$, $B(\cdot)$ are appropriate real-valued functions. The SGD procedures simplify to

$$\boldsymbol{\theta}_n^{\mathrm{sgd}} = \boldsymbol{\theta}_{n-1}^{\mathrm{sgd}} + \boldsymbol{C}_n(\boldsymbol{s}(\boldsymbol{y}_n) - \nabla A(\boldsymbol{\theta}_{n-1}^{\mathrm{sgd}})), \tag{13}$$

$$\boldsymbol{\theta}_n^{\mathrm{im}} = \boldsymbol{\theta}_{n-1}^{\mathrm{im}} + \boldsymbol{C}_n(\boldsymbol{s}(\boldsymbol{y}_n) - \nabla A(\boldsymbol{\theta}_n^{\mathrm{im}})). \quad (14)$$

In what follows, we will consider a *frequentist* evaluation of SGD as a statistical estimation method i.e., we will consider $\boldsymbol{\theta}_n^{\mathrm{sgd}}$ (or $\boldsymbol{\theta}_n^{\mathrm{im}}$ ) to be an *estimator* of $\theta_\bigstar$, and we will focus on its bias and variance across multiple realizations of the data set $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n\}$, under the same model and parameter $\theta_\bigstar$.[4]

### 3.1 Asymptotic bias and variance

Typically, online procedures such as SGD have two phases, namely the *ex- ploration phase* (or search phase) and the *convergence phase* [Amari, 1998, Benveniste et al., 2012]. In the exploration phase the iterates rapidly approach $\theta_\bigstar$, whereas in the convergence phase they jitter around $\theta_\bigstar$ within a ball of slowly decreasing radius. We will overview a typical analysis of SGD in the final convergence phase in which we assume that a Taylor approximation in the neighborhood of $\theta_\bigstar$ is accurate [Murata, 1998, Toulis et al., 2014]. In particular let $\boldsymbol{\mu}(\theta) = \mathbb{E}(\nabla \ell(\theta; \boldsymbol{y}_n))$, and assume that

$$\boldsymbol{\mu}(\boldsymbol{\theta}_n) = \boldsymbol{\mu}(\boldsymbol{\theta}_\star) + \boldsymbol{J}_\mu(\boldsymbol{\theta}_\star)(\boldsymbol{\theta}_n - \boldsymbol{\theta}_\star) + o(a_n), \quad (15)$$

where $\boldsymbol{J}_\mu$ is the Jacobian of the function $\boldsymbol{\mu}(\cdot)$, and $o(a_n)$ denotes a vector sequence $\boldsymbol{v}_n$ for which $\|\boldsymbol{v}_n\|/a_n \to 0$. Under typical regularity conditions $\boldsymbol{\mu}(\theta_\bigstar) = \boldsymbol{0}$ and $\boldsymbol{J}_\mu(\theta_\bigstar) = -\mathcal{I}(\theta_\bigstar)$. Thus, if we denote the biases of the two SGD methods as $\mathbb{E}(\boldsymbol{\theta}_n^{\mathrm{sgd}} - \boldsymbol{\theta}_\star) \triangleq b_n^{\mathrm{sgd}}$ and $\mathbb{E}(\boldsymbol{\theta}_n^{\mathrm{im}} - \boldsymbol{\theta}_\star) \triangleq b_n^{\mathrm{im}}$, by taking expectations in Equations (10) and (11) we obtain

$$\boldsymbol{b}_n^{\mathrm{sgd}} = (\boldsymbol{I} - \boldsymbol{C}_n \boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_\star))\, \boldsymbol{b}_{n-1}^{\mathrm{sgd}} + o(a_n), \quad (16)$$

$$\boldsymbol{b}_n^{\mathrm{im}} = (\boldsymbol{I} + \boldsymbol{C}_n \boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_\star))^{-1}\, \boldsymbol{b}_{n-1}^{\mathrm{im}} + o(a_n). \quad (17)$$

We observe that the convergence rate at which the two methods become unbiased in the limit differ in two significant ways. First, the explicit SGD method converges faster than the implicit one because $\|(\boldsymbol{I} - \boldsymbol{C}_n\, \mathcal{I}(\theta_\bigstar))\| < \|(\boldsymbol{I} + \boldsymbol{C}_n\, \mathcal{I}(\theta_\bigstar))^{-1}\|$, for sufficiently large $n$; 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 condition matrices $\boldsymbol{C}_n$. For example, the explicit SGD requires that the sequence $\boldsymbol{I} - \boldsymbol{C}_n\, \mathcal{I}(\theta_\bigstar)$ is comprised of matrices with eigenvalues less than one, in order to guarantee stability; this is a significant source of trouble when applying explicit SGD in practice. In contrast, for *any* specification of positive-definite $\boldsymbol{C}_n$, the eigenvalues of $(\boldsymbol{I} + \boldsymbol{C}_n\, \mathcal{I}(\theta_\bigstar))^{-1}$ are less than one, and thus implicit SGD is *unconditionally stable*; we will discuss more about stability in Section 3.4.

[4] This is an important distinction because, traditionally, the focus in optimization has been to obtain fast convergence to some point $\hat{\theta}$ that minimizes the empirical loss e.g., the maximum-likelihood estimator. From a statistical viewpoint, under variability of the data, there is a trade-off between convergence to an estimator and its asymptotic variance [Le Cun and Bottou, 2004].

In regard to statistical efficiency, Taylor approximation can also be used to establish recursive equations for the asymptotic variance of $\boldsymbol{\theta}_n^{\mathrm{sgd}}$ and $\boldsymbol{\theta}_n^{\mathrm{im}}$. For example, Toulis et al. [2014] show that if $\boldsymbol{C}$ is a symmetric matrix that commutes with $\mathcal{I}(\theta_\bigstar)$ such that $(2\boldsymbol{C}\,\mathcal{I}(\theta_\bigstar)-\boldsymbol{I})$ is positive-definite and $n\boldsymbol{C}_n \rightarrow \boldsymbol{C}$, it holds

$$\begin{aligned} n\mathrm{Var}(\boldsymbol{\theta}_n^{\mathrm{sgd}}) &\rightarrow (2\boldsymbol{C}\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)-\boldsymbol{I})^{-1}\boldsymbol{C}\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)\boldsymbol{C}^\top, \\ n\mathrm{Var}(\boldsymbol{\theta}_n^{\mathrm{im}}) &\rightarrow (2\boldsymbol{C}\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)-\boldsymbol{I})^{-1}\boldsymbol{C}\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)\boldsymbol{C}^\top; \end{aligned} \quad (18)$$

i.e., both SGD methods have the same asymptotic variance. Thus, for firstorder SGD procedures where $\boldsymbol{C}_n = a_n\boldsymbol{I}$ with $na_n \rightarrow \alpha > 0$ we obtain

$$\begin{aligned} n\mathrm{Var}(\boldsymbol{\theta}_n^{\mathrm{sgd}}) &\rightarrow \alpha^2(2\alpha\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)-\boldsymbol{I})^{-1}\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star), \\ n\mathrm{Var}(\boldsymbol{\theta}_n^{\mathrm{im}}) &\rightarrow \alpha^2(2\alpha\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)-\boldsymbol{I})^{-1}\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star). \end{aligned} \quad (19)$$

The matrix term $(2\boldsymbol{C}\,\mathcal{I}(\theta_\bigstar)-\boldsymbol{I})^{-1}$ represents the information that is lost by SGD, and it needs to be identity for optimal statistical efficiency (see Section 3.4). In fact, in more generality, this term is equal to $(2\boldsymbol{C}\boldsymbol{J}_\mu(\theta_\bigstar)-\boldsymbol{I})^{-1}$ where $\boldsymbol{\mu}(\theta)$ is mean value function of the statistic used in SGD (see also Equation (15)), and $\boldsymbol{J}_\mu(\theta_\bigstar)$ is its Jacobian at the true parameter values. Therefore, the asymptotic efficiency of SGD methods depends crucially on the Jacobian of the mean value function of the statistic used in the SGD iterations.

Asymptotic variance results similar to (18) were first studied in the stochastic approximation literature by Chung [1954], Sacks [1958], and followed by Fabian [1968b] and several other authors [see also Ljung et al., 1992, Parts I, II], but not in a closed-form (18), as most analyses were not done under the context of recursive statistical estimation. Furthermore, Sakrison's asymptotic efficiency result [Sakrison, 1965] can be recovered by setting $\boldsymbol{C}_n = (1/n)\,\mathcal{I}(\theta_{n-1})^{-1}$; in this case the asymptotic variance for both estimators is $(1/n)\,\mathcal{I}(\theta_\bigstar)^{-1}$ i.e., it is the optimal asymptotic efficiency of the maximum likelihood estimator.

## 3.2 Stability issues

Stability has been a well-known issue for explicit SGD. The main problem in practice is that the learning rate sequence needs to agree with the eigenvalues of the Fisher information matrix. To see this, let us simplify (16) and (17) by dropping the remainder terms $o(a_n)$. Then we obtain

$$\boldsymbol{b}_n^{\mathrm{sgd}}=(\boldsymbol{I}-\boldsymbol{C}_n\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star))\boldsymbol{b}_{n-1}^{\mathrm{sgd}}=\boldsymbol{P}_1^n\boldsymbol{b}_0, \quad (20)$$

$$\boldsymbol{b}_n^{\mathrm{im}}=(\boldsymbol{I}+\boldsymbol{C}_n\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star))^{-1}\boldsymbol{b}_{n-1}^{\mathrm{im}}=\boldsymbol{Q}_1^n\boldsymbol{b}_0, \quad (21)$$

where $\boldsymbol{P}_1^n=\prod_{i=1}^n(\boldsymbol{I}-\boldsymbol{C}_i\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)), \boldsymbol{Q}_n^1=\prod_{i=1}^n(\boldsymbol{I}+\boldsymbol{C}_i\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star))^{-1}$, and $\boldsymbol{b}_0$ denotes the initial bias of the two procedures from some common starting point $\theta_0$. Thus, the matrices $\boldsymbol{P}_1^n$ and $\boldsymbol{Q}_1^n$ describe how fast the initial bias decays for the explicit and implicit SGD respectively.

Assuming convergence, $\boldsymbol{P}_1^n \to \mathbf{0}$ and $\boldsymbol{Q}_1^n \to \mathbf{0}$ , and thus we say that both methods are *asymptotically stable*. However, they have significant differences in small-to-moderate samples. For simplicity, let us compare the two SGD procedures in their first-order formulation where $\boldsymbol{C}_n = a_n\boldsymbol{I}$ and $a_n = \alpha/n$ for some $\alpha > 0$.

In explicit SGD, the eigenvalues of $\boldsymbol{P}_1^n$ can be calculated as $\lambda_i'=\prod_j(1-\alpha\lambda_i/j)=\mathcal{O}(n^{-\alpha\lambda_i})$, for $0 < \alpha\lambda_i < 1$, where $\lambda_i$ are the eigenvalues of the Fisher information matrix $\mathcal{I}(\theta_\star)$. Thus, the magnitude of $\boldsymbol{P}_1^n$ 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 $\mathcal{I}(\theta_\star)$. The condition $\alpha\lambda_{max} \leq 1 \Rightarrow \alpha \leq 1/\lambda_{max}$ is required for stability, but for fast convergence we require $\alpha\lambda_{min} \approx 1$. In high-dimensional settings, this could be the source of serious problems because $\lambda_{max}$ could be at the order of $p$ i.e., the number of model parameters. Thus, in explicit SGD the requirements for stability and speed of convergence are in conflict. A conservative learning rate sequence can guarantee stability but this comes at a price in convergence which will be at the order of $\mathcal{O}(n^{-\alpha\lambda}\text{min})$, and vice versa. In stark contrast, the implicit procedure is *unconditionally stable*. The eigenvalues of $\boldsymbol{Q}_1^n$ are $\lambda_i'=\prod_{j=1}^n 1/(1+\alpha\lambda_i/j)=\mathcal{O}(n^{-\alpha\lambda_i})$, and thus are guaranteed to be less than one for any choice of the learning rate parameter $\alpha$. The critical difference with explicit SGD is that it is no longer required to have a small $\alpha$ for stability because the eigenvalues of $\boldsymbol{Q}_1^n$ will always be less than one.

Based on this analysis the magnitude of $\boldsymbol{P}_1^n$ can become arbitrarily large, and thus explicit SGD is likely to numerically diverge. In contrast, $\boldsymbol{Q}_1^n$ is guaranteed to be bounded, and so under any misspecification of the learning rate parameter the implicit SGD procedure is guaranteed to remain stable. The instability of explicit SGD is well-known, and requires careful work to be avoided in practice. For example, a typical learning rate for explicit SGD is of the form $a_n = \alpha(\alpha\beta+n)^{-1}$, where $\beta$ is chosen so that the explicit updates will not diverge; a reasonable choice is to set $\beta = \text{trace}(\mathcal{I}(\theta_\star))$ and $\alpha$ to be set close to $1/\lambda_{min}$. Such *explicit* normalization of the learning rates is not necessary in implicit SGD because, as shown in Equation (4), the implicit update performs such normalization indirectly.

Finally, an important line of work in the stability of stochastic approximations has been inspired by Huber's work in robust statistics [Huber et al., 1964, Huber, 2011]. In our notation, robust stochastic approximation considers iterations of the following form

$$\boldsymbol{\theta}_n=\boldsymbol{\theta}_{n-1}+\boldsymbol{C}_n\psi(\boldsymbol{s}(\boldsymbol{y}_n)-\boldsymbol{h}(\boldsymbol{\theta}_{n-1})), \quad (22)$$

where an appropriate function $\psi$ is sought for robust estimation; in this problem we assume $\mathbb{E}(\boldsymbol{s}(\boldsymbol{y}_n)) = h(\theta_\star)$ but the distribution of $\boldsymbol{s}(\boldsymbol{y}_n) - \boldsymbol{h}(\theta_\star)$ – denoted by $f(\cdot)$ – is unknown. In a typical setup, $f(\cdot)$ is considered to belong to a family of distributions $\mathcal{P}$, and $\psi$ is selected as

$$\boldsymbol{\psi}_\star=\arg_\psi \min \max_{f\in\mathcal{P}} \lim_{n\to\infty} n\mathrm{Var}(\boldsymbol{\theta}_n)$$

i.e., such that the maximum possible variance over the family $\mathcal{P}$ is minimized. Several important results have been achieved by Martin and Masreliez [1975] and Polyak and Tsypkin [1979]. For example, in linear models where $\boldsymbol{\mu}(\cdot)$ is linear in $\theta$ and $\boldsymbol{s}(\boldsymbol{y}_n)$ is one-dimensional, consider the general family $\mathcal{P} = \{f: f(0) \geq \varepsilon\}$ as the set of all symmetric densities that are positive at 0. Then the optimal choice is $\psi_\bigstar = \text{sign}(\cdot)$ i.e., the sign function, because it can be shown that the Laplace distribution is the member density of $\mathcal{P}$ that gives the least information about the parameters $\theta_\bigstar$.

### 3.3 Choice of parameterization and efficiency

First-order SGD methods are attractive for their computational performance, but the variance result (19) shows that they may suffer a significant loss in statistical efficiency. However, a reparameterization of the problem could yield a first-order SGD method that is optimal. The method can be described as follows. First, assume the exponential family (12) such that $\nabla\ell(\theta; \boldsymbol{y}_n) = \boldsymbol{s}(\boldsymbol{y}_n) - \boldsymbol{h}(\theta)$, where $h(\theta) = \nabla A(\theta) = \mathbb{E}(\boldsymbol{s}(\boldsymbol{y}_n)| \ \theta_\bigstar = \theta)$, and consider the reparameterization

$$\boldsymbol{\omega} \triangleq \boldsymbol{h}(\boldsymbol{\theta}), \quad (23)$$

which we assume it exists, it is 1-1 and easy to compute; these are critical assumptions that are hard, but not impossible to hold in practice. We will refer to (23) as the *mean-value parameterization* and $\omega$ as the mean-value parameters. Starting with an estimate $\omega_0$ of $\omega_\bigstar = \boldsymbol{h}(\theta_\bigstar)$, we can define the SGD procedures on this new parameter space as

$$\boldsymbol{\omega}_n^{\text{sgd}} = \boldsymbol{\omega}_{n-1}^{\text{sgd}} + (1/n)(\boldsymbol{s}(\boldsymbol{y}_n) - \boldsymbol{\omega}_{n-1}^{\text{sgd}}), \quad (24)$$

$$\boldsymbol{\omega}_n^{\text{im}} = \boldsymbol{\omega}_{n-1}^{\text{im}} + (1/n)(\boldsymbol{s}(\boldsymbol{y}_n) - \boldsymbol{\omega}_n^{\text{im}}), \quad (25)$$

where we also set $\boldsymbol{C}_n = (1/n)\boldsymbol{I}$ so that $\boldsymbol{C} = \boldsymbol{I}$. In this case, the explicit SGD simply calculates the running average of the complete sufficient statistic i.e., $\boldsymbol{\omega}_n^{\text{sgd}} = n^{-1}\sum_{i=1}^n \boldsymbol{s}(\boldsymbol{y}_i)$, and thus it is identical to the MLE estimator; similarly the implicit SGD satisfies $\boldsymbol{\omega}_n^{\text{im}} = (n+1)^{-1}\sum_{i=1}^n \boldsymbol{s}(\boldsymbol{y}_i)$ i.e., it is a slightly biased version of the MLE. It is thus straightforward to show (see for example Toulis and Airoldi [2014]) that the mean-value parameterization is optimal i.e.,

$$\text{Var}\,(\boldsymbol{h}^{-1}(\boldsymbol{\omega}_n^{\text{sgd}})) \rightarrow (1/n)\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)^{-1}, \text{Var}\,(\boldsymbol{h}^{-1}(\boldsymbol{\omega}_n^{\text{im}})) \rightarrow (1/n)\boldsymbol{\mathscr{I}}(\boldsymbol{\theta}_\star)^{-1}. \quad (26)$$

Intuitively, the mean-value parameterization transforms all parameters into location parameters. The Jacobian of the regression function of the statistic is $\boldsymbol{J}_\mu(\omega_\bigstar) = \nabla_\omega \mathbb{E}(\boldsymbol{s}(\boldsymbol{y}_n)| \ \omega = \omega_\bigstar) = \boldsymbol{I}$, and thus the information loss described in Equation (18) is avoided since $(2\boldsymbol{C}\boldsymbol{J}_\mu(\omega_\bigstar) - \boldsymbol{I})^{-1} = \boldsymbol{I}$. Transforming back to the original parameter space incurs no information loss as well, and so estimation of $\theta_\bigstar$ is efficient. This method is illustrated in the following example.

*Example*. Consider the problem of estimating ($\mu$, $\sigma^2$) from normal observations $y_n \sim \mathcal{N}(\mu, \sigma^2)$, and let $\boldsymbol{\theta}_\bigstar = (\mu, \sigma^2)$ which is not the natural parameterization. Consider sufficient statistics $\boldsymbol{s}(\boldsymbol{y}_n) = (y_n, y_n^2)$ such that $\mathbb{E}(\boldsymbol{s}(\boldsymbol{y}_n)) = (\mu, \mu^2 + \sigma^2) \triangleq (\omega_1, \omega_2)$. The parameter $\boldsymbol{\omega} = (\omega_1, \omega_2)$ corresponds to the mean-value parameterization. The inverse transformation is $\mu = \omega_1$ and $\sigma^2 = \omega_2 - \omega_1^2$, and thus its Jacobian is

$$\boldsymbol{J}_h^{-1} = \begin{pmatrix} 1 & 0 \\ -2\omega_1 & 1 \end{pmatrix}.$$

The variance of $\boldsymbol{s}(\boldsymbol{y}_n)$ is given by

$$\boldsymbol{V}(\boldsymbol{\theta}_\star) = \begin{pmatrix} Q & 2\omega_1 Q \\ 2\omega_1 Q & 4\omega_1^2 Q + 2Q^2 \end{pmatrix},$$

where $Q = \omega_2 - \omega_1^2 = \sigma^2$. Thus the variance of ($\hat{\omega}_1, \hat{\omega}_2$) is $(1/n)\boldsymbol{V}(\boldsymbol{\theta}_\bigstar)$ and the variance of $(\mu, \widehat{\sigma^2})$ is given by

$$\operatorname{Var}\left((\hat{\mu}, \hat{\sigma^2})\right) = (1/n)\boldsymbol{J}_h^{-1}\boldsymbol{V}\,\boldsymbol{J}_h^{-1\boldsymbol{T}} = (1/n)\begin{pmatrix} Q & 0 \\ 0 & 2Q^2 \end{pmatrix} = (1/n)\begin{pmatrix} \sigma^2 & 0 \\ 0 & 2\sigma^4 \end{pmatrix},$$

which is exactly the asymptotic variance of the MLE estimate. In practice, however, the mean-value transformation is rarely possible. Still, the intuition of transforming the model parameters into location parameters can be very useful in many situations, even when such transformation is approximate.

### 3.4 Choice of learning rate sequence

An interesting observation on the asymptotic variance results (18) is that for any choice of the symmetric positive-definite matrix $\boldsymbol{C}$,

$$(2\boldsymbol{C}\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_\star) - \boldsymbol{I})^{-1}\boldsymbol{C}\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_\star)\boldsymbol{C}^\top \geq \boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_\star)^{-1}, \quad (27)$$

where $\boldsymbol{A} \geq \boldsymbol{B}$ for two matrices $\boldsymbol{A}, \boldsymbol{B}$ indicates that $\boldsymbol{A} - \boldsymbol{B}$ is nonnegative-definite. Even in second-order form, both methods incur an efficiency loss when compared to the maximum-likelihood estimator, which can be quantified exactly through (18). Thus, there are two ways to achieve asymptotic efficiency.

First, one can design the condition matrix such that $n\boldsymbol{C}_n \rightarrow \mathcal{I}(\boldsymbol{\theta}_\bigstar)^{-1} \triangleq \boldsymbol{C}_\bigstar$.[5] However, this requires knowledge of the Fisher information matrix on the true parameters $\boldsymbol{\theta}_\bigstar$, which is

[5] Similarly, a sequence of matrices $\boldsymbol{C}_n$ can be designed such that $\boldsymbol{C}_n \rightarrow \mathcal{I}(\boldsymbol{\theta}_\bigstar)^{-1}$ [Sakrison, 1965].

usually unknown. The Venter process [Venter, 1967] was the first method to follow an adaptive approach to estimate this 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 $\boldsymbol{C}_\star$ (e.g., through a Quasi-Newton scheme) have recently been applied with considerable success [Schraudolph et al., 2007, Bordes et al., 2009]; see Section 3.5.2 for more details.

In contrast, an efficiency loss is generally unavoidable in first-order SGD i.e., when $\boldsymbol{C}_n = a_n \boldsymbol{I}$ with $na_n \to \alpha$. Asymptotic efficiency can occur only when $\lambda_i = 1/\alpha$ i.e., when all eigenvalues $\lambda_i$ of the Fisher information matrix $\mathcal{I}(\theta_\star)$ are identical. When $\lambda_i$'s are distinct the eigenvalues of the asymptotic variance matrix $n\mathrm{Var}(\boldsymbol{\theta}_n^{\mathrm{sgd}})$ (or $n\mathrm{Var}(\boldsymbol{\theta}_n^{\mathrm{im}})$) are $\alpha^2\lambda_i/(2\alpha\lambda_i - 1)$ which is at least $1/\lambda_i$ for any $\alpha$. In this case, one reasonable way to set the parameter $\alpha$ would be to minimize the trace of the asymptotic variance matrix i.e., solve

$$\hat{\alpha} = \arg\min_{\alpha} \sum_i \alpha^2 \lambda_i / (2\alpha\lambda_i - 1), \quad (28)$$

under the constraint that $\alpha > 1/(2\lambda_{min})$, thus making an undesirable but necessary comprise for convergence in all parameter components. However, the eigenvalues $\{\lambda_i\}$ 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]. A powerful alternative is to *reparametrize* the problem, apply SGD on the new parameter space, and then perform the inverse transformation, as in Section 3.3.

**3.4.1 Practical considerations—**There is voluminous amount of research literature on learning rate sequences for stochastic approximation and SGD. However, we decided to discuss this issue at the end of this section because the choice of the learning rate sequence conflates multiple design goals that are usually conflicting in practice e.g., convergence (or bias), asymptotic variance, stability and so on.

In general, the theory presented so far indicates that the learning rate for first-order explicit SGD should be of the form $a_n = \alpha(\alpha\beta + n)^{-1}$. Note that $\lim_{n\to\infty} na_n = \alpha$, so $\alpha$ is indeed the learning rate parameter introduced in Section 1. Parameter $\alpha$ will control the asymptotic variance and a reasonable choice would be the solution of (28), which requires estimates of the eigenvalues of the Fisher information matrix $\mathcal{I}(\theta_\star)$. An easier method is to simply 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 for (28), and also has good empirical performance [Xu, 2011, Toulis et al., 2014]. Parameter $\beta$ can be used to stabilize explicit SGD.

In particular, one would want to control the variance of the stochastic gradient Var $(\nabla\ell(\theta_n; \boldsymbol{y}_n)) = \mathcal{I}(\theta_\star) + \mathcal{O}(a_n)$, for points near $\theta_\star$; see also the stability analysis in Section 3.2. One reasonable value would thus be $\beta = \mathrm{trace}(\mathcal{I}(\theta_\star))$, which can be estimated easily by summing norms of the score function i.e, $\hat{\beta} = \sum_{i=1}^{n} \|\nabla\ell(\boldsymbol{\theta}_i; \boldsymbol{y}_i)\|^2$, similar to [Amari et al., 2000, Duchi et al., 2011] – see also Section (3.5.2).

For implicit SGD, the situation is a bit easier because a learning rate sequence $a_n = \alpha(\alpha + n)^{-1}$ works well in practice [Toulis et al., 2014]. As before, $\alpha$ controls the efficiency of the method and so we can set $\alpha = 1/\lambda_{min}$ as in explicit SGD. The additional stability term ($\beta$) in explicit SGD is unnecessary because the implicit method performs such normalization (shrinkage) indirectly – see Equation (4).

However, tuning the learning rate sequence eventually depends on problem-specific considerations, and there is a considerable variety of sequences that have been employed in practice [George and Powell, 2006]. Principled design of learning rates in SGD remains an important research topic [Schaul et al., 2012].

## 3.5 Some interesting extensions

**3.5.1 Averaged stochastic gradient descent—**Estimation with SGD can be optimized for statistical efficiency only with knowledge of the underlying model. For example, the optimal learning rate parameter $\alpha$ in first-order SGD requires knowledge of the eigenvalues of the Fisher information matrix $\mathcal{I}(\theta_\star)$. In second-order SGD, optimality is achieved when one uses a sequence of matrices $\boldsymbol{C}_n$ such that $n\boldsymbol{C}_n \to \mathcal{I}(\theta_\star)^{-1}$. Methods that approximate $\mathcal{I}(\theta_\star)$ make up a significant class of methods in stochastic approximation. Another important class of stochastic approximation methods relies on *averaging* of the iterates. The corresponding SGD procedure is usually referred to as *averaged SGD*, or *ASGD* for short.[6]

Averaging of iterates in the Robbins-Monro procedures was studied independently by Ruppert [1988] and Bather [1989], and both proposed similar averaging schemes. If we use the notation of Section 2 (see also iteration (6)), Ruppert [1988] considered the following stochastic approximation procedure

$$\begin{aligned} \theta_n &= \theta_{n-1} - a_n y_n, \\ \bar{\theta}_n &= \frac{1}{n}\sum_{i=1}^{n} \theta_i, \end{aligned} \qquad (29)$$

where $a_n = \alpha n^{-c}$ for $1/2 < c < 1$ and $\bar{\theta}_n$ are the estimates of the zero of the regression function $M(\theta)$. Under certain conditions, Ruppert [1988] showed that $n\text{Var}(\bar{\theta}_n) \to \sigma^2/M'(\theta_\star)^2$, where $\sigma^2 = \text{Var}(y_n|\theta_\star)$. Recall, that the typical Robbins-Monro procedure gives estimates with asymptotic variance $\alpha^2\sigma^2/(2\alpha M'(\theta_\star) - 1)$, which is at least equal to the variance of the averaged iterate. Ruppert [1988] provides a nice statistical intuition on why averaging gives such efficiency with larger learning rates. First, write $y_n = M(\theta_n) - \varepsilon_n$, where $\varepsilon_n$ are zero-mean independent random variables with finite variance. The typical analysis in stochastic approximation starts by solving the recursion (6) to get an expression like the following

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

$$\theta_n - \theta_\star = \sum_{i=1}^{n} c(i,n) a_i \varepsilon_i + o(1), \quad (30)$$

where $c(i, n) = \exp\{-A(n) + A(i)\}$, $A(m)=K\sum_{j=1}^{m} a_j$ is the function of partial sums, and $K$ is some constant. Ruppert [1988] shows that Equation (30) can be rewritten as

$$\theta_n - \theta_\star = a_n \sum_{i=b(n)}^{n} c(i,n) \varepsilon_i + o(1), \quad (31)$$

where $b(n) = \lfloor n - Rn^c \log n \rfloor$ with $R$ a positive constant, and $\lfloor \cdot \rfloor$ the positive integer floor function. Ruppert [1988] argues that when $a_n = \alpha/n$ then $b(n) = \mathcal{O}(1)$, and $\theta_n - \theta_\bigstar$ is the weighted average of all noise variables $\varepsilon_n$. When $a_n = \alpha n^{-c}$ for $1/2 < c < 1$, then $\theta_n - \theta_\bigstar$ is a weighted average of only $\mathcal{O}(n^c \log n)$ noise variables. Thus, in the former case there is significant autocorrelation in the series $\theta_n$. In the latter case, for $0 < p_1 < p_2 < 1$ the variables $\theta_{\lfloor p_1 n \rfloor}$ and $\theta_{\lfloor p_2 n \rfloor}$ are asymptotically uncorrelated, and thus averaging improves the 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 mild assumptions. In fact, due to the authors' prior work in averaged stochastic approximation, ASGD is usually referred to as *Polyak-Ruppert* averaging scheme. Adoption of averaging schemes for statistical learning has been slow but steady over the years [Zhang, 2004, Nemirovski et al., 2009, Bottou, 2010, Cappé, 2011]. One practical reason is that averaging only helps when when the underlying stochastic process is slow to converge, which is hard to know in practice; in fact, averaging can have an adverse effect when the underlying SGD process is converging well. Furthermore, the selection of the learning rate sequence is also important in ASGD, and a bad sequence can cause the algorithm to converge very slowly [Xu, 2011], or even diverge. Research on ASGD is still ongoing as several directions, such as the combination of stable methods with averaging schemes, remain unexplored (e.g., stochastic proximal methods, implicit SGD). Furthermore, in a similar line of work several methods have been developed that use averaging in order to reduce the variance of stochastic gradients [Johnson and Zhang, 2013, Wang et al., 2013].

**3.5.2 Second-order stochastic gradient descent—**Sakrison's recursive estimation method (9) is the archetype of second-order SGD, but it requires an expensive matrix inversion at every iteration. Several methods have been developed that approximate such a matrix across iterations in stochastic approximation, and are generally termed *adaptive*. Early adaptive methods in stochastic approximation were given by Nevelson and Khasminskĭ [1973] and Wei [1987]; translated into a SGD procedure, such methods would recursively estimate $\mathcal{I}(\theta_\bigstar)$ by computing finite-differences $\boldsymbol{y}_{n,+}^{j} - \boldsymbol{y}_{n,-}^{j}$ sampled at $\theta_n + c_n \boldsymbol{e}_j$ and $\theta_n - c_n \boldsymbol{e}_j$ respectively, where $\boldsymbol{e}_j$ is the $j$th unit basis vector and $c_n$ is an appropriate sequence of positive numbers. While such methods are very useful in sequential experiment design where one has control over the data generation process, they are impractical for modern online learning problems.

A simple and effective approach was proposed by Amari et al. [2000]. The idea is to keep an estimate $\hat{\mathcal{I}}_n$ of $\mathcal{I}(\theta_\star)$ and use an explicit SGD scheme as follows:

$$\begin{aligned} \hat{\mathcal{I}}_n &= (1-c_n)\hat{\mathcal{I}}_{n-1} + c_n \nabla\ell(\boldsymbol{\theta}_{n-1};\boldsymbol{y}_n)\nabla\ell(\boldsymbol{\theta}_{n-1};\boldsymbol{y}_n)^\top, \\ \boldsymbol{\theta}_n &= \boldsymbol{\theta}_{n-1} + \hat{\mathcal{I}}_n^{-1}\nabla\ell(\boldsymbol{\theta}_{n-1};\boldsymbol{y}_n). \end{aligned} \qquad (32)$$

Inversion of the estimate $\hat{\mathcal{I}}_n$ is (relatively) cheap by using the Sherman-Morrison formula. This scheme, however, introduces the additional problem of determining the sequence $c_n$ in (32). In their work, Amari et al. [2000] advocated for a small costant $c_n = c > 0$ that can be determined through computer simulations.

Another notable approach based on Quasi-Newton methods (see Section 1) was developed by Bordes et al. [2009]. Their method, termed *SGD-QN*, approximates the Fisher information matrix through a secant condition as in the original BFGS algorithm [Broyden, 1965]. The secant condition in SGD-QN is

$$\boldsymbol{\theta}_n - \boldsymbol{\theta}_{n-1} \approx \hat{\mathcal{I}}_{n-1}^{-1}[\nabla\ell(\boldsymbol{\theta}_n;\boldsymbol{y}_n) - \nabla\ell(\boldsymbol{\theta}_{n-1};\boldsymbol{y}_n)] \triangleq \hat{\mathcal{I}}_{n-1}^{-1}\boldsymbol{\delta}_n, \qquad (33)$$

where $\hat{\mathcal{I}}_n$ is kept diagonal. If we let $\boldsymbol{D}_n$ denote the diagonal matrix with $i$th diagonal element $d_{ii} = (\theta_{n,i} - \theta_{n-1,i})/\delta_{n,i}$, then the update of the approximation matrix in SGD-QN is given by

$$\hat{\mathcal{I}}_n \leftarrow \hat{\mathcal{I}}_{n-1} + \frac{2}{r}(\boldsymbol{D}_n - \hat{\mathcal{I}}_{n-1}), \qquad (34)$$

and the update of $\theta_n$ is similar to (32). The parameter $r$ is controlled internally in the algorithm, and counts the number of times the update (34) has been performed.

A notable second-order method is also *AdaGrad* [Duchi et al., 2011], which adapts multiple learning rates using gradient information. In one popular variant of the method, AdaGrad keeps a diagonal ($p \times p$) matrix $\boldsymbol{A}_n$ of learning rates that is updated at every iteration. Upon observing data $\boldsymbol{y}_n$, AdaGrad updates $\boldsymbol{A}_n$ as follows:

$$\boldsymbol{A}_n = \boldsymbol{A}_{n-1} + \mathrm{diag}(\nabla\ell(\boldsymbol{\theta}_{n-1};\boldsymbol{y}_n)\nabla\ell(\boldsymbol{\theta}_{n-1};\boldsymbol{y}_n)^\top), \qquad (35)$$

where diag($\boldsymbol{A}$) is the diagonal matrix with the same diagonal as its matrix argument $\boldsymbol{A}$. Learning in AdaGrad proceeds through the iteration

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \alpha \boldsymbol{A}_n^{-1/2} \circ \nabla\ell(\boldsymbol{\theta}_{n-1};\boldsymbol{y}_n), \qquad (36)$$

where $\alpha > 0$ is a learning rate parameter that is shared among all parameter components, and the symbol ∘ denotes element wise multiplication. The original motivation for AdaGrad stems from proximal methods in optimization, but there is a statistical intuition why the update (36) is reasonable. In general, from an information perspective, a learning rate sequence $a_n$ discounts an observation $\boldsymbol{y}_n$ according to the reciprocal of the *statistical information* that has been gathered so far for the parameter of interest $\boldsymbol{\theta}_\star$. The intuition behind a rate of the form $a_n = \alpha/n$ is that the information after $n$ iterations is proportional to $n$, under the i.i.d. data assumption. In many dimensions where some parameter component affects outcomes less frequently than others, AdaGrad replaces the term $n$ with an *estimate* of the information that has *actually* been received for that component. A (biased) estimate of this information is provided by the elements of $\boldsymbol{A}_n$ in (36), and is justified since $\mathbb{E}(\nabla\ell(\boldsymbol{\theta}; \boldsymbol{y}_n) \nabla\ell(\boldsymbol{\theta}; \boldsymbol{y}_n)^\top) = \mathcal{I}(\boldsymbol{\theta})$. Interestingly, implicit SGD and AdaGrad share the common property of shrinking explicit SGD estimates according to the Fisher information matrix. Second-order implicit SGD methods are yet to be explored, but further connections are possible.

**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, in many situations that are important in practice, this is not possible, for example when the likelihood is only known up to a normalizing constant. In such cases, definitions (10) and (11) cannot be applied directly since $\nabla\ell(\boldsymbol{\theta}; \boldsymbol{y}_n)$ cannot be computed. However, if unbiased samples of the log-likelihood gradients are available, then explicit SGD can be readily applied. This is possible if sampling from the model is relatively easy.

In particular, assume an exponential family model (12) that is easy to sample from e.g., through Metropolis-Hastings. A variant of explicit SGD, termed *Monte-Carlo* SGD [Toulis and Airoldi, 2014], can be constructed as follows. Starting from some estimate $\boldsymbol{\theta}_0^{\text{mc}}$ , iterate the following steps for each $n$th data point $\boldsymbol{y}_n$, where $n = 1, 2, \ldots N$:

1. Get $m$ samples from the model $\tilde{\boldsymbol{y}}_i \sim f(\cdot; \boldsymbol{\theta}_{n-1}^{\text{mc}})$, $i = 1, 2, \ldots m$.
2. Compute average sufficient statistic $\widetilde{\boldsymbol{s}_n} = (1/m)\sum_{i=1}^{m} \boldsymbol{s}(\widetilde{\boldsymbol{y}_i})$.
3. Update the estimate through

$$\boldsymbol{\theta}_n^{\text{mc}} = \boldsymbol{\theta}_{n-1}^{\text{mc}} + \boldsymbol{C}_n(\boldsymbol{s}(\boldsymbol{y}_n) - \widetilde{\boldsymbol{s}_n}). \quad (37)$$

The main idea of a Monte-Carlo SGD algorithm (37) is to use the current parameter estimate in order to *impute* the expected value of the sufficient statistic that would otherwise be available if the likelihood was easy to evaluate. Furthermore, assuming $n\boldsymbol{C}_n \to \boldsymbol{C}$, the asymptotic variance of the estimate satisfies

$$n\text{Var}\left(\boldsymbol{\theta}_n^{\text{mc}}\right) \to (1+1/m) \cdot (2\boldsymbol{C}\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_\star) - \boldsymbol{I})^{-1} \boldsymbol{C}\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_\star)\boldsymbol{C}^\top, \quad (38)$$

which exceeds the variance of the typical explicit SGD estimator by a factor of (1 + 1/$m$). However, in its current form the Monte-Carlo SGD (37) is only explicit; an implicit version would require to sample data from the next iterate, which is technically challenging but an interesting open problem. Still, an approximate implicit implementation of Monte-Carlo SGD is possible using the intuition in Equation (4). For example, one could simply run an explicit update as in (37), but then shrink according to $(\boldsymbol{I} + a_n\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_n^{\text{mc}}))^{-1}$, or more efficiently using a one-dimensional shrinkage factor $(1+a_n \mathrm{trace}(\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}_n^{\mathrm{mc}})))^{-1}$, for some decreasing sequence $a_n > 0$.

Theoretically Monte-Carlo SGD is based on *sampling-controlled* stochastic approximation methods in which the usual regression function of the Robbins-Monro procedure (6) is only accessible through sampling [Dupuis and Simha, 1991] e.g., through MCMC. Convergence in such settings is subtle because it also depends on the ergodicity of the underlying Markov chain [Younes, 1999]. In practice, approximate variants of the aforementioned Monte-Carlo SGD procedure have been applied with considerable success to fit large models of neural networks, notably through the contrastive divergence algorithm, as we briefly discuss in Section 4.4.

# 4 Selected applications

SGD has found several important applications over the years. In this section we will review some of them, giving a preference to breadth over depth.

## 4.1 Online EM algorithm

The Expectation-Maximization algorithm [Dempster et al., 1977] is a numerically stable procedure to compute the maximum likelihood estimator in latent variable models. Extending our notation, let $\boldsymbol{x}_n$ denote a latent variable at observed data point $\boldsymbol{y}_n$, and let $f_{\mathrm{com}}(\boldsymbol{x}_n, \boldsymbol{y}_n; \theta)$ and $f_{\mathrm{obs}}(\boldsymbol{y}_n; \theta)$ denote the complete-data and observed-data density, respectively; similarly, $\ell_{\mathrm{com}}$ and $\ell_{\mathrm{obs}}$ will denote the respective log-likelihoods. For simplicity, we will assume that $f_{\mathrm{com}}$ is an exponential family model in the natural parameterization, as in (12), such that

$$f_{\mathrm{com}}(\boldsymbol{x}_n, \boldsymbol{y}_n; \boldsymbol{\theta}) = \exp\{\boldsymbol{s}(\boldsymbol{x}_n, \boldsymbol{y}_n)^\top \boldsymbol{\theta} - A(\boldsymbol{\theta}) + B(\boldsymbol{x}_n, \boldsymbol{y}_n)\}. \quad (39)$$

We will denote the corresponding Fisher information matrices as $\mathcal{I}_{\mathrm{com}}(\theta) = -\mathbb{E}(\nabla\nabla\ell_{\mathrm{com}}(\boldsymbol{x}_n, \boldsymbol{y}_n; \theta))$ and $\mathcal{I}_{\mathrm{obs}}(\theta) = \mathbb{E}(\nabla\nabla\ell_{\mathrm{obs}}(\boldsymbol{y}_n; \theta))$, where the expectations are considered with model parameters fixed at $\theta$. Furthermore, let $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$ denote the entire observed data set as in Section 1, and $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ be the corresponding latent variables. The traditional EM algorithm proceeds by iterating the following steps.

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n; \boldsymbol{Y}) = \mathbb{E}\left(\ell_{\mathrm{com}}(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{\theta}) | \boldsymbol{\theta}_n, \boldsymbol{Y}\right), \qquad E\text{−}step \quad (40)$$

$$\boldsymbol{\theta}_{n+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n; \boldsymbol{Y}). \qquad \mathbf{M{-}step} \quad (41)$$

Dempster et al. [1977] showed that the EM algorithm converges to the maximum-likelihood estimator $\hat{\theta} = \mathrm{argmax}_\theta\, \ell_{\mathrm{obs}}(\boldsymbol{Y}; \theta)$; furthermore, they showed that EM is an ascent algorithm i.e., the likelihood is strictly increasing at each iteration, and thus EM has a desirable numerical stability. However, the EM algorithm is impractical for the analysis of large data sets because it involves expensive operations, both in the expectation and maximization steps, that need to be performed on the entire data set. Therefore, online schemes are necessary for analysis of large models with latent variables.

Titterington [1984] considered a procedure defined through the iteration

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + a_n \mathscr{I}_{\text{com}}(\boldsymbol{\theta}_{n-1})^{-1} \nabla \ell_{\text{obs}}(\boldsymbol{y}_n; \boldsymbol{\theta}_{n-1}). \quad (42)$$

This procedure differs only marginally from Sakrison’s recursive estimation method (see Section 2.2) by using the complete-data information matrix. In the univariate case where the true model parameter is $\theta_\bigstar$, Titterington [1984] applied Fabian’s theorem [Fabian, 1968a] to show that the estimate in (42) satisfies $\sqrt{n}(\theta_n - \theta_\star) \sim \mathscr{N}(0, \mathscr{I}_{\text{com}}(\theta_\star))^{-2} \mathscr{I}_{\text{obs}}(\theta_\star)/(2\mathscr{I}_{\text{obs}}(\theta_\star)\mathscr{I}_{\text{com}}(\theta_\star)^{-1} - 1)$. Thus, as in the traditional full-data EM algorithm, the efficiency of the online method (42) depends on the amount of missing information. Notably, Lange [1995] considered Newton-Raphson iterations for the M-step of the EM algorithm, and derived an online procedure that is similar to (42).

However, procedure (42) is essentially an explicit stochastic gradient method, and thus it may have serious stability and convergence problems, contrary to the desirable numerical stability of EM. In the exponential family model (39), Nowlan [1991] considered one of the first “true” online EM algorithms as follows:

$$\begin{aligned} \boldsymbol{s}_{n+1} &= \boldsymbol{s}_n + \alpha \mathbb{E}\left(\boldsymbol{s}(\boldsymbol{x}_n, \boldsymbol{y}_n; \boldsymbol{\theta}_n) | \boldsymbol{\theta}_n, \boldsymbol{y}_n\right), \quad && E-step \\ \boldsymbol{\theta}_{n+1} &= \arg\max_{\boldsymbol{\theta}} \ell_{\text{com}}(\boldsymbol{s}_{n+1}; \boldsymbol{\theta}), && M-step \end{aligned} \quad (43)$$

where $\alpha \in (0, 1)$. In words, the algorithm starts from some initial sufficient statistic $\boldsymbol{s}_0$ and then updates it through a stochastic approximation scheme with a constant step size $\alpha$. The maximization step is identical to that of traditional EM. Online EM with decreasing step sizes was later developed by Sato and Ishii [2000] as follows:

$$\begin{aligned} \boldsymbol{s}_{n+1} &= \boldsymbol{s}_n + \alpha_n \left[\mathbb{E}\left(\boldsymbol{s}(\boldsymbol{x}_n, \boldsymbol{y}_n; \boldsymbol{\theta}_n) | \boldsymbol{\theta}_n, \boldsymbol{y}_n\right) - s_n\right], \quad && E-step \\ \boldsymbol{\theta}_{n+1} &= \arg\max_{\boldsymbol{\theta}} \ell_{\text{com}}(\boldsymbol{s}_{n+1}; \boldsymbol{\theta}). && M-step \end{aligned} \quad (44)$$

By the theory of stochastic approximation, procedure (44) will converge to the observed-data maximum likelihood estimate $\hat{\theta}$. In contrast, procedure (43) will not converge with a constant $\alpha$, but it will reach a point in the vicinity of $\hat{\theta}$ more rapidly than (44). Further extensions of the aforementioned 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 [Neal and Hinton, 1998, Sato and Ishii, 2000, Liu et al., 2006, Cappé, 2011].

### 4.2 MCMC sampling

Let $\theta$ be the model parameters of observations $\boldsymbol{Y} = (\boldsymbol{y}_1, \cdots \boldsymbol{y}_N)$, with an assumed prior distribution denoted by $\pi(\theta)$. A common task in Bayesian statistics it to sample from the posterior distribution $f(\theta|\boldsymbol{Y}) \propto \pi(\theta)f(\boldsymbol{Y}|\theta)$. The Hamiltonian Monte-Carlo (HMC) [Neal, 2011] is a method in which auxiliary variables $\boldsymbol{p}$ are introduced to the original variables $\theta$ to improve sampling from $f(\theta|\boldsymbol{Y})$. In the augmented parameter space, we consider a function $H(\theta, \boldsymbol{p}) = U(\theta) + K(\boldsymbol{p}) \in \mathbb{R}^+$, where $U(\theta) = -\log f(\theta|\boldsymbol{Y})$ and $K(\boldsymbol{p}) = (1/2)\boldsymbol{p}^\top \boldsymbol{M}\boldsymbol{p}$ with a symmetric positive-definite matrix $\boldsymbol{M}$. Next, we consider the density

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

In this parameterization, the variables $\boldsymbol{p}$ are independent of $\theta$. Assuming some initial state $(\theta_0, \boldsymbol{p}_0)$, HMC sampling proceeds in iterations indexed by $n = 1, \cdots$, as follows:

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

Step 2. is the key idea in HMC. The variables $(\theta, \boldsymbol{p})$ can be mapped to a physical system where $\theta$ is the *position* of the system, and $\boldsymbol{p}$ is the *momentum*. The Hamiltonian dynamics refer to a set of ordinary differential equations (ODE) that govern the movement of the system, and thus calculate the future values of $(\theta, \boldsymbol{p})$ given a pair of current values. Being a closed physical system, the *Hamiltonian* of the system is constant. Thus, in Step 3. of HMC it holds $-H(\theta_n, \boldsymbol{p}_n) + H(\theta_{n-1}, \boldsymbol{p}^*) = 0$, and thus the acceptance probability is one.

A special case of HMC, called *Langevin dynamics*, defines the sampling iterations as follows [Girolami and Calderhead, 2011]:

$$\begin{gathered}\boldsymbol{\eta}_n \sim \mathcal{N}(\mathbf{0}, \varepsilon \boldsymbol{I}),\\ \boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \tfrac{\varepsilon}{2}(\nabla \log \pi(\boldsymbol{\theta}_{n-1}) + \log f(\boldsymbol{\theta}_{n-1}; \boldsymbol{Y})) + \boldsymbol{\eta}_n.\end{gathered} \quad (45)$$

The sampling procedure (45) follows from HMC by a numerical solution of the ODE method in Step 2. of the algorithm using the *leapfrog method*. Parameter $\varepsilon > 0$ 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 (45) using a stochastic gradient as follows:

$$\begin{gathered}\boldsymbol{\eta}_n \sim \mathcal{N}(0, \varepsilon_n),\\ \boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \tfrac{\varepsilon_n}{2}(\nabla \log \pi(\boldsymbol{\theta}_{n-1}) + (N/b) \sum_{i \in \text{batch}} \nabla \log f(\boldsymbol{y}_i|\boldsymbol{\theta}_{n-1})) + \boldsymbol{\eta}_n.\end{gathered} \quad (46)$$

The step-sizes $\varepsilon_n$ satisfy the typical requirements in stochastic approximation i.e., $\Sigma \varepsilon_i = \infty$ and $\sum \varepsilon_i^2 < \infty$. Procedure (46) is using stochastic gradients averaged over a *mini-batch* of $b$ samples that is usually employed in SGD to reduce noise in the stochastic gradients. Notably, Sato and Nakagawa [2014] proved that procedure (46) converges to the true posterior $f(\theta|\boldsymbol{Y})$ with an elegant use of stochastic calculus. Sampling through stochastic gradient Langevin dynamics has since generated a lot of significant work in MCMC sampling for very large data sets, and it is still a rapidly expanding research area with contributions from various disciplines [Hoffman et al., 2013, Pillai and Smith, 2014, Korattikara et al., 2014].

### 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 $\boldsymbol{x}_t \in \mathcal{X}$, where $\mathcal{X}$ is some state-space. Upon entering a state $\boldsymbol{x}_t$ two things happen. First, an agent receives a probabilistic *reward* $R(\boldsymbol{x}_t) \in \mathbb{R}$, and then 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 $\pi$: $\mathcal{X} \to \mathcal{A}$, thus mapping a state to an action. Nature then decides a *transition* to state $\boldsymbol{x}_{t+1}$ through a density $p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)$ that is unknown to the agent.

One important task in reinforcement learning is to estimate the *value function* $V^{\pi}(\boldsymbol{x})$ which quantifies the expected value of a specific state $\boldsymbol{x} \in \mathcal{X}$ for an agent. This is defined as

$$V^{\pi}(\boldsymbol{x})=\mathbb{E}\left(R(\boldsymbol{x})\right)+\gamma\mathbb{E}\left(R(\boldsymbol{x}_1)\right)+\gamma^2\mathbb{E}\left(R(\boldsymbol{x}_2)\right)+\ldots, \quad (47)$$

where $\boldsymbol{x}_t$ denotes the state that will be reached starting at $\boldsymbol{x}$ after $t$ transitions, and $\gamma \in (0, 1)$ is a parameter that discounts future rewards. Note that the variation of $R(\boldsymbol{x}_t)$ includes the uncertainty of the state $\boldsymbol{x}_t$ because of the stochasticity in transitions, and the uncertainty from the reward distribution. Thus, $V^{\pi}(\boldsymbol{x})$ admits a recursive definition as follows:

$$V^{\pi}(\boldsymbol{x})=\mathbb{E}\left(R(\boldsymbol{x})\right)+\gamma\mathbb{E}\left(V^{\pi}(\boldsymbol{x}_1)\right). \quad (48)$$

When the state is a high-dimensional vector, one popular approach is to use the *linear value approximation* $V(\boldsymbol{x}) = \boldsymbol{\theta}_{\bigstar}{}^{\top}\varphi(\boldsymbol{x})$, where $\varphi(\boldsymbol{x})$ maps a state to *features* in a space with fewer dimensions, and $\boldsymbol{\theta}_{\bigstar}$ is a vector of fixed parameters. If an agent is at state $\boldsymbol{x}_t$, then the recursive equation (48) can be rewritten as

$$\mathbb{E}\left(R(\boldsymbol{x}_t)-(\boldsymbol{\theta}_{\star}{}^{\top}\boldsymbol{\phi}_t-\gamma\boldsymbol{\theta}_{\star}{}^{\top}\boldsymbol{\phi}_{t+1})|\boldsymbol{\phi}_t\right)=0, \quad (49)$$

where we set $\varphi_t = \varphi(\boldsymbol{x}_t)$ for notational convenience. Similar to SGD, this suggests a stochastic approximation method to estimate $\boldsymbol{\theta}_{\bigstar}$ through the following iteration:

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+a_t\left[R(\boldsymbol{x}_t)-(\boldsymbol{\theta}_t^{\top}\boldsymbol{\phi}_t-\gamma\boldsymbol{\theta}_t^{\top}\boldsymbol{\phi}_{t+1})\right]\boldsymbol{\phi}_t, \quad (50)$$

where $a_t$ is a learning rate sequence that satisfies the Robbins-Monro conditions (see Section 2.1). Equation (50) is known as the *temporal differences* (TD) learning algorithm [Sutton, 1988]. Implicit versions of this algorithm have recently emerged in order to solve some of the known stability issues of the classical TD algorithm [Schapire and Warmuth, 1996, Li, 2008, Wang and Bertsekas, 2013, Tamar et al., 2014]. For example, Tamar et al. [2014] consider computing the term $\boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t$ at the future iterate, and thus the resulting *implicit* TD algorithm is

$$\boldsymbol{\theta}_{t+1} = (\boldsymbol{I} + a_t \boldsymbol{\phi}_t \boldsymbol{\phi}_t^\top)^{-1} [\boldsymbol{\theta}_t + a_t (R(\boldsymbol{x}_t) + \gamma \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1}) \boldsymbol{\phi}_t]. \quad (51)$$

Similar to implicit SGD, iteration (51) stabilizes the TD iterations. With the advent of online multiagent markets, methods and applications in reinforcement learning have been receiving a renewed stream of research effort [Gosavi, 2009].

### 4.4 Deep learning

Deep learning is the task of estimating parameters of statistical models that can be represented by multiple layers of non-linear 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, in most situations the models are 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 an observation ($\boldsymbol{x}$, $\boldsymbol{y}$) of the states of hidden and observed input units respectively, is given by

$$P(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \frac{\exp\{-\boldsymbol{b}' \boldsymbol{y} - \boldsymbol{c}' \boldsymbol{x} - \boldsymbol{x}' \boldsymbol{W} \boldsymbol{y}\}}{Z(\boldsymbol{\theta})}, \quad (52)$$

where $\theta = (\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{W})$ are the model parameters, and the normalizing constant is $Z(\theta) = \Sigma_{\boldsymbol{x},\boldsymbol{y}} \exp\{-\boldsymbol{b}'\boldsymbol{y} - \boldsymbol{c}'\boldsymbol{x} - \boldsymbol{x}'\boldsymbol{W}\boldsymbol{y}\}$ (also known as the partition function). Furthermore, the sample spaces for $\boldsymbol{x}$ and $\boldsymbol{y}$ are discrete (e.g., binary) and finite. The observed-data density is thus $P(\boldsymbol{y}; \theta) = \Sigma_{\boldsymbol{x}} P(\boldsymbol{x}, \boldsymbol{y}; \theta)$. Let $H(\boldsymbol{x}, \boldsymbol{y}; \theta) = \boldsymbol{b}'\boldsymbol{y} + \boldsymbol{c}'\boldsymbol{x} + \boldsymbol{x}'\boldsymbol{W}\boldsymbol{y}$, such that $P(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \frac{e^{-H(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})}}{Z(\boldsymbol{\theta})}$. Through simple algebra one can obtain the log-likelihood of an observed sample $\boldsymbol{y}^{\text{obs}}$ in the following convenient form:

$$\nabla \ell(\boldsymbol{\theta}; \boldsymbol{y}^{\text{obs}}) = -[\mathbb{E}(\nabla H(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})) - \mathbb{E}(\nabla H(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) | \boldsymbol{y}^{\text{obs}})]. \quad (53)$$

In practical situations the data $\boldsymbol{x}$, $\boldsymbol{y}$ are binary. Therefore the conditional distribution of the missing data $\boldsymbol{x}|\boldsymbol{y}$ is readily available through the usual logistic regression GLM model, and thus the second term of (53) is easy to sample from. Similarly, $\boldsymbol{y}|\boldsymbol{x}$ is easy to sample from. However, the first term in (53) requires sampling from the joint distribution of the complete data ($\boldsymbol{x}$, $\boldsymbol{y}$), which conceptually is easy to sample from using the aforementioned conditionals and a Gibbs sampling scheme [Geman and Geman, 1984]. However, the state space for both $\boldsymbol{x}$ and $\boldsymbol{y}$ is usually very large e.g., comprised of thousands or millions of units, and thus a full Gibbs on the joint distribution is usually 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 $\boldsymbol{y}^{(i)}$ from the empirical distribution of observed states.
2. Sample $\boldsymbol{x}^{(i)}|\boldsymbol{y}^{(i)}$ i.e., the hidden state.
3. Sample $\boldsymbol{y}^{(i,new)}|\boldsymbol{x}^{(i)}$.
4. Sample $\boldsymbol{x}^{(i,new)}|\boldsymbol{y}^{(i,new)}$.
5. Evaluate the gradient (53) using $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$ for the second term, and the sample $(\boldsymbol{x}^{(i,new)}, \boldsymbol{y}^{(i,new)})$ for the first term.
6. Update the parameters in $\theta$ using constant-step size SGD and the estimated gradient from Step 4.

In other words, contrastive divergence estimates both terms of (53). This estimation is biased because $(\boldsymbol{x}^{(i,new)}, \boldsymbol{y}^{(i,new)})$ is assumed to be from the full joint distribution of $(\boldsymbol{x}, \boldsymbol{y})$. In fact, contrastive divergence might operate in $k$ steps in which the Steps 3–4 are repeated $k$ times, in an effort to approximate the joint distribution better by letting the chain run longer. Although in theory larger $k$ should approximate the full joint better, it has been 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].

## Acknowledgments

The authors wish to thank Leon Bottou, Bob Carpenter, David Dunson, Andrew Gelman, Brian Kulis, Xiao-Li Meng, Natesh Pillai, Neil Shephard, Daniel Sussman and Alexander Volfovsky for useful comments and discussion. This research was sponsored, in part, by NSF CAREER award IIS-1149662, ARO MURI awardW911NF-11-1-0036, and ONR YIP award N00014-14-1-0485. PT is a Google Fellow in Statistics. EMA is an Alfred P. Sloan Research Fellow.

## References

Amari, Shun-Ichi. Natural gradient works efficiently in learning. Neural computation. 1998; 10(2): 251–276.

Amari, Shun-Ichi; Park, Hyeyoung; Fukumizu, Kenji. Adaptive method of realizing natural gradient learning for multilayer perceptrons. Neural Computation. 2000; 12(6):1399–1409. [PubMed: 10935719]

Bather, JA. Stochastic approximation: A generalisation of the Robbins-Monro procedure. Vol. 89. Mathematical Sciences Institute, Cornell University; 1989.

Beck, Amir; Teboulle, Marc. Mirror descent and nonlinear projected subgradient methods for convex optimization. Operations Research Letters. 2003; 31 (3):167–175.

Bengio, Yoshua. Learning deep architectures for ai. Foundations and trends® in Machine Learning. 2009; 2(1):1–127.

Bengio, Yoshua; Delalleau, Olivier. Justifying and generalizing contrastive divergence. Neural Computation. 2009; 21(6):1601–1621. [PubMed: 19018704]

Benveniste, Albert; Métivier, Michel; Priouret, Pierre. Adaptive algorithms and stochastic approximations. Springer Publishing Company, Incorporated; 2012.

Bertsekas, Dimitri P.; Tsitsiklis, John N. Neuro-dynamic programming: an overview. Decision and Control, 1995., Proceedings of the 34th IEEE Conference on; IEEE; 1995. p. 560-564.

Bordes, Antoine; Bottou, Léon; Gallinari, Patrick. Sgd-qn: Careful quasi-newton stochastic gradient descent. The Journal of Machine Learning Research. 2009; 10:1737–1754.

Bottou, Léon. Large-scale machine learning with stochastic gradient descent. Proceedings of COMPSTAT'2010; Springer; 2010. p. 177-186.

Bottou, Léon; Le Cun, Yann. On-line learning for very large data sets. Applied Stochastic Models in Business and Industry. 2005; 21(2):137–151.

Bousquet, Olivier; Bottou, Léon. The tradeoffs of large scale learning. Advances in neural information processing systems. 2008:161–168.

Broyden, Charles G. A class of methods for solving nonlinear simultaneous equations. Mathematics of computation. 1965:577–593.

Cappé, Olivier. Online em algorithm for hidden markov models. Journal of Computational and Graphical Statistics. 2011; 20(3)

Cappé, Olivier; Moulines, Eric. On-line expectation–maximization algorithm for latent data models. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2009; 71(3):593–613.

Carreira-Perpinan, Miguel A.; Hinton, Geoffrey E. On contrastive divergence learning. Proceedings of the tenth international workshop on artificial intelligence and statistics; Citeseer. 2005. p. 33-40.

Chung, Kai Lai. On a stochastic approximation method. The Annals of Mathematical Statistics. 1954:463–483.

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

Duchi, John; Hazan, Elad; Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. The Journal of Machine Learning Research. 2011; 999999:2121–2159.

Dupuis, Paul; Simha, Rahul. On sampling controlled stochastic approximation. Automatic Control, IEEE Transactions on. 1991; 36(8):915–924.

Fabian, Vaclav. On asymptotic normality in stochastic approximation. The Annals of Mathematical Statistics. 1968a:1327–1332.

Fabian, Vaclav. On asymptotic normality in stochastic approximation. The Annals of Mathematical Statistics. 1968b:1327–1332.

Fabian, Vaclav. Asymptotically efficient stochastic approximation; the rm case. The Annals of Statistics. 1973:486–495.

Fisher, RA. Statistical Methods for Research Workers. Oliver and Boyd; Edinburgh: 1925a.

Fisher, Ronald A. 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. 1922; 222:309–368.

Fisher, Ronald Aylmer. Theory of statistical estimation. Mathematical Proceedings of the Cambridge Philosophical Society; Cambridge Univ Press; 1925b. p. 700-725.

Geman, Stuart; Geman, Donald. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 1984; 6:721–741.

George, Abraham P.; Powell, Warren B. Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. Machine learning. 2006; 65(1):167–198.

Girolami, Mark; Calderhead, Ben. Riemann manifold langevin and hamiltonian monte carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2011; 73(2):123–214.

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

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

Hastie, T.; Tibshirani, R.; Friedman, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2. Springer; 2011.

Hennig, Philipp; Kiefel, Martin. Quasi-newton methods: A new direction. The Journal of Machine Learning Research. 2013; 14(1):843–865.

Hinton, Geoffrey E. Training products of experts by minimizing contrastive divergence. Neural computation. 2002; 14(8):1771–1800. [PubMed: 12180402]

Hoffman, Matthew D.; Blei, David M.; Wang, Chong; Paisley, John. Stochastic variational inference. The Journal of Machine Learning Research. 2013; 14(1): 1303–1347.

Huber, Peter J. Robust statistics. Springer; 2011.

Huber, Peter J., et al. Robust estimation of a location parameter. The Annals of Mathematical Statistics. 1964; 35(1):73–101.

Johnson, Rie; Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. Advances in Neural Information Processing Systems. 2013:315–323.

El Karoui, Noureddine. Spectrum estimation for large dimensional covariance matrices using random matrix theory. The Annals of Statistics. 2008:2757–2790.

Kivinen, Jyrki; Warmuth, Manfred K. Additive versus exponentiated gradient updates for linear prediction. Proceedings of the twenty-seventh annual ACM symposium on Theory of computing; ACM; 1995. p. 209-218.

Kivinen, Jyrki; Warmuth, Manfred K.; Hassibi, Babak. The p-norm generalization of the lms algorithm for adaptive filtering. Signal Processing, IEEE Transactions on. 2006; 54(5):1782–1793.

Korattikara, Anoop; Chen, Yutian; Welling, Max. Austerity in mcmc land: Cutting the metropolis-hastings budget. Proceedings of The 31st International Conference on Machine Learning; 2014. p. 181-189.

Kulis, Brian; Bartlett, Peter L. Implicit online learning. Proceedings of the 27th International Conference on Machine Learning (ICML-10); 2010. p. 575-582.

Lai TL, Robbins Herbert. Adaptive design and stochastic approximation. The annals of Statistics. 1979:1196–1221.

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

Lange, Kenneth. Numerical analysis for statisticians. Springer; 2010.

Bottou, Leon; Le Cun, Yann; Bottou, L. Large scale online learning. Advances in neural information processing systems. 2004; 16:217.

Lehmann, EH.; Casella, G. Theory of Point Estimation. 2. Springer; 2003.

Li, Lihong. A worst-case comparison between temporal difference and residual gradient with linear function approximation. Proceedings of the 25th international conference on machine learning; ACM; 2008. p. 560-567.

Liu, Zikuan; Almhana, Jalal; Choulakian, Vartan; McGorman, Robert. Online em algorithm for mixture with application to internet traffic modeling. Computational statistics & data analysis. 2006; 50(4):1052–1071.

Ljung, Lennart; Pflug, Georg; Walk, Harro. Stochastic approximation and optimization of random systems. Vol. 17. Springer; 1992.

Douglas Martin R, Masreliez C. Robust estimation via stochastic approximation. Information Theory, IEEE Transactions on. 1975; 21(3):263–271.

Murata, Noboru. Online Learning and Neural Networks. Cambridge University Press; Cambridge, UK: 1998. A statistical study of on-line learning.

Nagumo, Jin-Ichi; Noda, Atsuhiko. A learning method for system identification. Automatic Control, IEEE Transactions on. 1967; 12(3):282–287.

National Research Council. Frontiers in Massive Data Analysis. The National Academies Press; Washington, DC: 2013.

Neal, Radford. Mcmc using hamiltonian dynamics. Handbook of Markov Chain Monte Carlo. 2011; 2

Neal, Radford M.; Hinton, Geoffrey E. Learning in graphical models. Springer; 1998. A view of the em algorithm that justifies incremental, sparse, and other variants; p. 355-368.

Nemirovski, Arkadi; Juditsky, Anatoli; Lan, Guanghui; Shapiro, Alexander. Robust stochastic approximation approach to stochastic programming. SIAM Journal on Optimization. 2009; 19(4): 1574–1609.

Nemirovski, DB.; Yudin. Problem complexity and method efficiency in optimization. Wiley; Chichester and New York: 1983.

Nevelson, Mikhail Borisovich; Khasminskĭ, Rafail Zalmanovich. Stochastic approximation and recursive estimation. Vol. 47. Amer Mathematical Society; 1973.

Nowlan, Steven J. Soft competitive adaptation: neural network learning algorithms based on fitting statistical mixtures. 1991.

Parikh, Neal; Boyd, Stephen. Proximal algorithms. Foundations and Trends in Optimization. 2013; 1(3):123–231.

Pillai, Natesh S.; Smith, Aaron. Ergodicity of approximate mcmc chains with applications to large data sets. 2014 arXiv preprint arXiv:1405.0182.

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

Polyak BT, Tsypkin YZ. Adaptive algorithms of estimation (convergence, optimality, stability). Avtomatika i Telemekhanika (Automatics and Remote Control). 1979:74–84.

Robbins, Herbert; Monro, Sutton. A stochastic approximation method. The Annals of Mathematical Statistics. 1951:400–407.

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

Rosasco, Lorenzo; Villa, Silvia; Vũ, Bang Công. Convergence of stochastic proximal gradient algorithm. 2014 arXiv preprint arXiv:1403.5074.

Ruppert, David. Technical report. Cornell University Operations Research and Industrial Engineering; 1988. Efficient estimations from a slowly convergent robbins-monro process.

Ryu, Ernest K.; Boyd, Stephen. Stochastic proximal iteration: A nonasymptotic improvement upon stochastic gradient descent.

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

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

Salakhutdinov, Ruslan; Mnih, Andriy; Hinton, Geoffrey. Restricted boltzmann machines for collaborative filtering. Proceedings of the 24th international conference on Machine learning; ACM; 2007. p. 791-798.

Sato, Issei; Nakagawa, Hiroshi. Approximation analysis of stochastic gradient langevin dynamics by using fokker-planck equation and ito process. JMLR W&CP. 2014; 32(1):982–990.

Sato, Masa-Aki; Ishii, Shin. On-line em algorithm for the normalized gaussian network. Neural computation. 2000; 12(2):407–432. [PubMed: 10636949]

Schapire, Robert E.; Warmuth, Manfred K. On the worst-case analysis of temporal-difference learning algorithms. Machine Learning. 1996; 22(1–3):95–121.

Schaul, Tom; Zhang, Sixin; LeCun, Yann. No more pesky learning rates. 2012 arXiv preprint arXiv: 1206.1106.

Schraudolph, Nicol; Yu, Jin; Günter, Simon. A stochastic quasi-newton method for online convex optimization. 2007

Li Cheng, SVN.; Schuurmans, Vishwanathan Dale; Caelli, Shaojun Wang Terry. Implicit online learning with kernels. Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference; MIT Press; 2007. p. 249

Slock, Dirk TM. On the convergence behavior of the lms and the normalized lms algorithms. Signal Processing, IEEE Transactions on. 1993; 41(9):2811–2825.

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

Tamar, Aviv; Toulis, Panos; Mannor, Shie; Airoldi, Edoardo. Implicit temporal differences. Neural Information Processing Systems, Workshop on large-scale reinforcement learning; 2014.

Taylor, Graham W.; Hinton, Geoffrey E.; Roweis, Sam T. Modeling human motion using binary latent variables. Advances in neural information processing systems. 2006:1345–1352.

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

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

Toulis, Panos; Airoldi, Edoardo M. Implicit stochastic gradient descent for principled estimation with large datasets. 2014 arXiv preprint arXiv:1408.2923.

Venter JH. An extension of the robbins-monro procedure. The Annals of Mathematical Statistics. 1967:181–190.

Wang, Chong; Chen, Xi; Smola, Alex; Xing, Eric. Variance reduction for stochastic gradient optimization. Advances in Neural Information Processing Systems. 2013:181–189.

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

Wei CZ. Multivariate adaptive stochastic approximation. The Annals of Statistics. 1987:1115–1130.

Welling, Max; Teh, Yee W. Bayesian learning via stochastic gradient langevin dynamics. Proceedings of the 28th International Conference on Machine Learning (ICML-11); 2011. p. 681-688.

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

Younes, Laurent. On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. Stochastics: An International Journal of Probability and Stochastic Processes. 1999; 65(3–4):177–228.

Zhang, Tong. Solving large scale linear prediction problems using stochastic gradient descent algorithms. Proceedings of the twenty-first international conference on Machine learning; ACM; 2004. p. 116