LETTER Communicated by Shun-ichi Amari

Adaptive Learning Algorithm Convergence in Passive and Reactive Environments

Richard M. Golden
golden@utdallas.edu
University of Texas at Dallas, Richardson, TX 75080, U.S.A.

Although the number of artificial neural network and machine learning architectures is growing at an exponential pace, more attention needs to be paid to theoretical guarantees of asymptotic convergence for novel, nonlinear, high-dimensional adaptive learning algorithms. When properly understood, such guarantees can guide the algorithm development and evaluation process and provide theoretical validation for a particular algorithm design. For many decades, the machine learning community has widely recognized the importance of stochastic approximation theory as a powerful tool for identifying explicit convergence conditions for adaptive learning machines. However, the verification of such conditions is challenging for multidisciplinary researchers not working in the area of stochastic approximation theory. For this reason, this letter presents a new stochastic approximation theorem for both passive and reactive learning environments with assumptions that are easily verifiable. The theorem is widely applicable to the analysis and design of important machine learning algorithms including deep learning algorithms with multiple strict local minimizers, Monte Carlo expectation-maximization algorithms, contrastive divergence learning in Markov fields, and policy gradient reinforcement learning.

1 Overview

Although the number of artificial neural network and machine learning architectures is growing at an exponential pace, more attention needs to be paid to theoretical guarantees of asymptotic convergence for novel, nonlinear, high-dimensional adaptive learning algorithms. When properly understood, such guarantees can guide the algorithm development and evaluation process and, in addition, provide theoretical validation for a particular algorithm design. For many decades, the machine learning community has widely recognized the importance of stochastic approximation theory as a powerful tool for identifying explicit convergence conditions for adaptive learning machines. However, the verification of such conditions is challenging for multidisciplinary researchers not working in the area of stochastic approximation theory. For this reason, the goal of this letter is to

Neural Computation **30, 2805–2832** (2018)
doi:10.1162/neco_a_01117

present a new stochastic approximation theorem with easily verifiable assumptions for characterizing the asymptotic behavior of a wide range of important machine learning algorithms.

The new stochastic approximation theorem presented here is applicable to the analysis of the asymptotic behavior of a wide range of learning algorithms including (1) deep learning algorithm (Bottou, 1991, 1998, 2004; Bengio, Courville, & Vincent, 2013; Sutskever, Marten, Dahl, & Hinton, 2013; Zhang, Choromanska, & LeCun, 2015), (2) variable metric (Jani, Dowling, Golden, & Wang, 2000; Paik, Golden, Torlak, & Dowling, 2006; Roux, Manzagol, & Bengio, 2008; Schraudolph, Yu, & Günter, 2007; Sunehag, Trumpf, Vishwanathan, & Schraudolph, 2009) and momentum-type stochastic approximation schemes (Pearlmutter, 1992; Roux, Schmidt, & Bach, 2012; Sutskever et al., 2013; Zhang et al., 2015), (3) reinforcement learning and adaptive control (Jaakkola, Jordan, & Singh, 1994; Baird & Moore, 1999; Williams, 1992; Sugiyama, 2015; Sutton & Barto, 1998; Balcan & Feldman, 2013; Mohri, Rostamizadeh, & Talwalkar, 2012), (4) expectation-maximization problems for latent variable and missing data problems (Carbonetto, King, & Hamze, 2009; Gu & Kong, 1998), and (5) contrastive divergence learning in Markov random fields (Yuille, 2005; Hinton, Osindero, & Teh, 2006; Tieleman, 2008; Swersky, Chen, Marlin, & de Freitas, 2010; Salakhutdinov & Hinton, 2012). A critical feature of the theorem is that its statement and proof are specifically designed to provide relatively easily verifiable assumptions and interpretable conclusions that can be understood and applied by researchers outside the field of stochastic approximation theory.

Stochastic approximation theorems have played a vital role in characterizing our understanding of adaptive learning algorithms from the very beginning of work in machine learning (e.g., Amari, 1967; Duda & Hart, 1973). White (1989a, 1989b), Benveniste, Metivier, and Priouret (1990), Bottou (1991), Bertsekas and Tsitsiklis (1996), Golden (1996), Borkar (2008), Swersky et al. (2010), and Mohri et al. (2012) provide useful discussions of the application of stochastic approximation methods to machine learning problems. Kushner (2010), a seminal contributor to the development of stochastic approximation theory, provides an excellent review of the theoretical stochastic approximation literature from its origins in the 1950s.

The generic form of a stochastic approximation algorithm is defined as follows. Consider a learning machine whose parameter values at iteration t of the learning algorithm are interpretable as the realization of a q-dimensional random vector $\tilde{\boldsymbol{\theta}}(t)$. The learning machine is provided an initial guess for the parameter estimates at iteration $t = 0$, which is denoted as $\tilde{\boldsymbol{\theta}}(0)$. Then the learning machine observes a realization of a random vector $\tilde{\mathbf{x}}(t)$ called the *training stimulus* $\mathbf{x}(t)$ which is then used to update the parameters of the learning machine.

In particular, the initial guess is then modified to obtain a revised parameter estimate at iteration $t + 1$, $\tilde{\boldsymbol{\theta}}(t + 1)$ using the formula

$$\tilde{\boldsymbol{\theta}}(t+1) = \tilde{\boldsymbol{\theta}}(t) + \gamma_t \tilde{\mathbf{d}}_t, \tag{1.1}$$

where the search direction is defined such that

$$\tilde{\mathbf{d}}_t \equiv \mathbf{d}_t(\tilde{\mathbf{x}}(t), \tilde{\boldsymbol{\theta}}(t))$$

and the step-size or learning rate γ_t is a positive number.

In the initial stages of learning, the *search time period*, the step-size γ_t is typically chosen to be either constant or to increase in value. During this phase of the learning process, the adaptive learning machine's dynamics in equation 1.1 have the opportunity to sample the statistical environment. Ideally, this time period should be sufficiently long so that there is an opportunity for the learning machine to observe the different types of training stimuli in its environment for the purpose of extracting critical statistical regularities. For example, if there are M distinct training stimuli that occur with approximately equal probability in the environment, then choosing the time period for learning to be $10M$ would ensure that each training stimulus will be approximately observed by the learning machine about 10 times during the initial search phase. After the initial search phase, the step-size γ_t is decreased at an appropriate rate to ensure convergence. This latter phase is called the *converge time period*.

Darken and Moody (1992) provide a good discussion of various types of search and converge strategies. For example, choosing

$$\gamma_t = \frac{\gamma_o\,((t/\tau)+1)}{(t/\tau)^2 + (t/\tau) + 1}, \tag{1.2}$$

where γ_0 is the initial positive step size and $t < \tau$ specifies the search time period where the step size is relatively constant, while $t \gg \tau$ corresponds to the "converge" time period where the step-size γ_t tends to decrease for $t = 0, 1, 2, \ldots$ Alternatively, one might choose the sequence of step-sizes $\gamma_1, \gamma_2, \ldots$ such that

$$\gamma_t = \frac{\gamma_o\,((t/\tau_1)+1)}{(t/\tau_2)^2 + 1}, \tag{1.3}$$

so that the step-size sequence initially increases for $t < \tau_1$ during the search phase and then decreases for $t \gg \tau_2 > \tau_1$ during the converge phase. Typically, the step size, γ_t, in equation 1.1 is a sequence of positive numbers that is relatively constant or increases in the initial stages of learning and then tends to decrease in the later stages of learning.

Different choices of the search direction vector $\tilde{\mathbf{d}}_t$ in equation 1.1 realize different popular stochastic descent algorithms such as stochastic gradient descent (Bottou, 1991, 1998), normalized stochastic gradient descent

(Hazan, Levy, & Shalev-Shwartz, 2015), modified Newton (Jani et al., 2000; Paik et al., 2006; Roux et al., 2008; Schraudolph et al., 2007; Sunehag et al., 2009), and momentum-type stochastic gradient descent methods (Pearlmutter, 1992; Roux et al., 2012; Sutskever et al., 2013; Zhang et al., 2015). A standard assumption is that the dot product of the expected value of the search direction $\tilde{\mathbf{d}}_t$ with the gradient of the objective function is less than or equal to zero.

Assume the stochastic sequence of d-dimensional random vectors $\tilde{\mathbf{x}}(1), \tilde{\mathbf{x}}(2), \ldots$ modeling the training stimuli are independent and identically distributed with common data generating process (DGP) probability density $p_e : \mathcal{R}^d \to [0, \infty)$. In other words, each time the learning machine updates its parameters, the likelihood of observing a particular training stimulus $\mathbf{x}(t)$ at iteration t is given by p_e. The goal of an adaptive learning machine is to estimate (learn) the global minimizer, $\boldsymbol{\theta}^* \in \mathcal{R}^q$, of a smooth risk function $\ell : \mathcal{R}^q \to \mathcal{R}$, which specifies the learning machine's optimal behavior. In addition, let a smooth function c be defined such that $c(\mathbf{x}, \boldsymbol{\theta})$ is the penalty, or "loss," incurred by the learning machine for choosing parameter value $\boldsymbol{\theta}$ for training stimulus $\mathbf{x}$ where $\mathbf{x} \in \mathcal{R}^d$.

In order to define the risk function in a general manner, let the notation

$$\int c(\mathbf{x}, \boldsymbol{\theta}) p_e(\mathbf{x}) d\nu(\mathbf{x})$$

denote $\sum c(\mathbf{x}, \boldsymbol{\theta}) p_e(\mathbf{x})$ when p_e is a probability mass function for a discrete random vector and $\int c(\mathbf{x}, \boldsymbol{\theta}) p_e(\mathbf{x}) d\mathbf{x}$ when p_e is an (absolutely continuous) probability density function for a continuous random vector. The notation $\int c(\mathbf{x}, \boldsymbol{\theta}) p_e(\mathbf{x}) d\nu(\mathbf{x})$ is also used to specify the appropriate combination of sums and Riemann integrals for common situations where the random training stimulus $\tilde{\mathbf{x}}$ includes both discrete and absolutely continuous random variables. Technically, p_e is a Radon-Nikodým density defined with respect to a sigma-finite measure ν, which additionally permits the representation of mixed random variables. The sigma-finite measure ν explicitly specifies which random variables are discrete and which are absolutely continuous. Mixed random variables that possess features of both absolutely continuous and discrete random variables can also be specified by the Radon-Nikodým density. For example, the setting of an analog volume control is not ideally modeled as either a discrete or continuous random variable since the probability that the volume control is set to the maximum value is positive, while the probability that the volume control is set to a value less than the maximum value is zero.

With this notation, the passive environment risk function ℓ is defined such that for all $\boldsymbol{\theta} \in \mathcal{R}^q$,

$$\ell(\boldsymbol{\theta}) = \int c(\mathbf{x}, \boldsymbol{\theta}) p_e(\mathbf{x}) d\nu(\mathbf{x}). \tag{1.4}$$

Thus, the goal of learning is to minimize the expected loss (or, equivalently, risk) associated with choosing a parameter value with respect to a particular statistical environment characterized by the DGP density p_e.

Several prior publications in the machine learning literature (White, 1989a, 1989b; Bottou, 1991, 1998; Golden, 1996; Mohri et al., 2012; Toulis, Rennie, & Airoldi, 2014) have provided explicit convergence theorems by considering parameter update equations of the form of equation 1.1 and assuming that the risk function has the form of equation 1.4. That is, at each parameter update, the training stimulus is sampled from the statistical environment using the probability density p_e. This assumption, unfortunately, is not directly relevant to many important problems in the areas of (1) contrastive divergence learning (Yuille, 2005; Younes, 1999; Hinton et al., 2006; Tieleman, 2008; Swersky et al., 2010; Salakhutdinov & Hinton, 2012); (2) learning in the presence of missing data or latent variables (Gu & Kong, 1998; Carbonetto et al., 2009; Vlassis & Toussaint, 2009); and (3) active learning and adaptive control (Jaakkola et al., 1994; Baird & Moore, 1999; Williams, 1992; Sugiyama, 2015; Sutton & Barto, 1998; Balcan & Feldman, 2013; Vlassis & Toussaint, 2009). Such problems typically require that the training stimulus is sampled from a statistical environment specified by the current parameter estimates so that rather than sampling from the density p_e, one samples from the density $p_e(\cdot|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the current knowledge state of the learning machine. These latter problems can be viewed as learning within a reactive learning environment.

Thus, rather than using the risk function in equation 1.4, the reactive learning environment risk function is defined such that for all $\boldsymbol{\theta} \in \mathcal{R}^q$,

$$\ell(\boldsymbol{\theta}) = \int c(\mathbf{x}, \boldsymbol{\theta}) p_e(\mathbf{x}|\boldsymbol{\theta}) d\nu(\mathbf{x}), \tag{1.5}$$

where the stochastic descent algorithm specified in equation 1.1 is based on the assumption that $\tilde{\mathbf{x}}(t)$ is sampled from a probability density function conditioned on the current state of the learning machine $\boldsymbol{\theta}(t)$. Thus, the learning environment's statistical characteristics are functionally dependent on the current knowledge state of the learning machine. In practice, this dependence is indirect since the environment is functionally dependent on the learning machine's behavior, which in turn is functionally dependent on the current knowledge state of the learning machine.

Note that a stochastic gradient descent algorithm minimizing the risk function in equation 1.5 can have quite a different functional form when compared with a stochastic gradient descent algorithm minimizing the risk function in equation 1.4. To see this, note the derivative of $\ell(\boldsymbol{\theta})$ in equation 1.4 for a passive statistical learning environment is specified by

$$\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \int \nabla_{\boldsymbol{\theta}} c(\mathbf{x}, \boldsymbol{\theta}) p_e(\mathbf{x}) d\nu(\mathbf{x}). \tag{1.6}$$

This formula for the gradient provided in equation 1.6 is not correct for a reactive learning environment where the risk function is given by equation 1.5. The gradient of equation 1.5 for a reactive statistical learning environment is given instead by the formula

$$\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \int \nabla_{\boldsymbol{\theta}} c(\mathbf{x}, \boldsymbol{\theta}) p_e(\mathbf{x}|\boldsymbol{\theta}) d\nu(\mathbf{x}) + \int c(\mathbf{x}, \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} p_e(\mathbf{x}|\boldsymbol{\theta}) d\nu(\mathbf{x}). \tag{1.7}$$

In the machine learning literature, most of the focus has been on investigating the rate of convergence of stochastic approximation algorithms (Roux et al., 2012; Mohri et al., 2012). Analyses in the machine learning literature (Yuille, 2005; Sunehag et al., 2009; Mohri et al., 2012) include theorems for handling reactive learning environments but do not explain in detail how such theorems handle the case where the data generating process density p_e is functionally dependent on $\boldsymbol{\theta}$ and do not explicitly characterize the asymptotic behavior of the state sequence $\{\tilde{\boldsymbol{\theta}}(t)\}$. In addition, such analyses often lack a discussion regarding how a stochastic approximation convergence theorem can be applied to situations where the objective function has multiple minimizers, maximizers, and saddle points. However, Blum (1954), Beneviste et al. (1990), Gu and Kong (1998), Kushner (1981), Younes (1999), and Delyon, Lavielle, & Moulines (1999) have provided explicit assumptions and proofs of convergence theorems for stochastic reactive learning environments, but the theorems and their assumptions may be difficult to apply in practice for readers without a background in stochastic approximation theory.

Clarity of understanding is important to ensure that such theorems can be properly and confidently applied in practice since the algorithms they describe are widely used in the field of machine learning. An important contribution of this letter is providing a relatively simple set of assumptions and a straightforward detailed discussion intended to support the mathematical analysis of a wide range of adaptive learning algorithms. Furthermore, it is hoped that as a result of the analyses presented here, the importance of prior contributions to the stochastic approximation theorem literature will be better appreciated and this analysis will serve as a stepping-stone to advanced study in this important area.

2 Overview of the New Convergence Theorem

The new stochastic approximation theorem that minimizes the reactive environment learning risk function in equation 1.5, as well as the passive learning risk function in equation 1.4, is similar to analyses by Andrieu, Moulines, and Priouret (2005), Blum (1954), Kushner (1981, theorem 1), White (1989a, 1989b), Benveniste et al. (1990; appendix to part II), Bertsekas and Tsitsiklis (1996, proposition 4.1, p. 141), Gu and Kong (1998), and Delyon et al. (1999, theorem 5). With respect to the machine learning literature,

the theorem and its proof are most closely related to the analysis of Sunehag et al. (2009). However, the assumptions, conclusions, and proof of this theorem are specifically designed to be easily understood by machine learning researchers working outside the field of stochastic approximation theory. The accessibility of these theoretical results is fundamentally important for the development of the field of machine learning to ensure that such results are correctly applied in specific applications. In addition to having conditions that are easily verifiable, the stochastic approximation theorem introduced here is applicable to a wide range of situations commonly encountered in practical machine learning problems.

If the objective function is positive definite everywhere on the parameter space, the theorem provides conditions ensuring convergence to the unique strict global minimum of the objective function. However, if the objective function has multiple minima, maxima, and saddle points, then the new stochastic approximation theorem is still applicable. In this latter nonconvex optimization case, the theorem provides the weaker conclusion that the sequence of algorithm-generated parameter estimates will converge to the set of critical points with probability one or the algorithm will generate a sequence of parameter estimates that are not bounded with probability one.

Note the terminology that an event occurs "with probability one" means there is a zero probability that the event will not occur. For example, if the stochastic sequence $\tilde{\boldsymbol{\theta}}(1), \tilde{\boldsymbol{\theta}}(2), \ldots$ converges to some set $\mathcal{H}$ with probability one, this means that the probability of observing any realization $\boldsymbol{\theta}(1), \boldsymbol{\theta}(2), \ldots$ that deterministically converges to $\mathcal{H}$ is exactly equal to one and the probability of observing any realization that does not converge to $\mathcal{H}$ is exactly equal to zero.

The theorem presented here assumes that the objective function is twice continuously differentiable. Although many important machine learning algorithms minimize objective functions that are not smooth, the points of discontinuity in the gradients of such objective functions can often be replaced with smooth transitions. For example, stochastic approximation methods are often used to minimize the objective function for a multilayer perceptron with rectified linear or "rectilinear" units (Glorot, Bordes, & Bengio, 2011; Zheng, Yang, Liu, Liang, & Li, 2015). Such an objective function is not continuously differentiable. Still, the theorem presented here is still relevant to the analysis of learning in such situations if one replaces the rectified linear units with softplus units (Glorot et al., 2011; Zheng et al., 2015), which are smooth approximations to rectified linear units. Note, in particular, that if one defines a more general form of the softplus transfer function, $\mathcal{S}(\phi) = \tau \log(1 + \exp(\phi/\tau))$, that for sufficiently large τ, the continuously differentiable softplus transfer function can approximate the nondifferentiable rectifier transfer function, $\mathcal{S}(\phi) = \max\{0, \phi\}$, as accurately as desired. Such situations are also typical in the context of important applications in machine learning involving L_1 regularization where the

nondifferentiable function penalty function $r(\phi) = |\phi|$ can be approximated with arbitrary precision with the differentiable penalty function,

$$r(\phi) = \tau \log(1 + \exp(\phi/\tau)) + \tau \log(1 + \exp(-\phi/\tau)),$$

for sufficiently large τ. For reactive learning environments when the objective function has the form of equation 1.5, the assumption that $\nabla \ell$ is continuous places smoothness constraints not only on the loss function c but also on the reactive data generating process specified by density $p_x(\cdot|\boldsymbol{\theta})$.

The assumption of a sufficiently smooth objective function is required for the analysis presented here for the purpose of showing that if $\nabla \ell(\tilde{\boldsymbol{\theta}}(t)) \to 0$ with probability one, then $\tilde{\boldsymbol{\theta}}(t)$ converges to a critical point with probability one. To see this, consider a simple squared-error type objective function $\ell : \mathcal{R} \to \mathcal{R}$ with L_1 regularization such as

$$\ell(\theta) \equiv (\theta - M)^2 + \lambda|\theta|,$$

where λ is a positive number. If $M \neq 0$, then even though ℓ is not differentiable everywhere, it is still possible to show that $\nabla \ell(\tilde{\theta}(t)) \to 0$ with probability one since the gradient of ℓ is defined in a neighborhood of a minimizer. However, when $M = 0$, then $\nabla \ell = 2\theta + \lambda$ for positive values of θ close to the global minimizer $\theta^* = 0$ and $\nabla \ell = 2\theta - \lambda$ for negative values of θ close to the global minimizer $\theta^* = 0$. At the point $\theta^* = 0$ when $M = 0$, the gradient of ℓ is not defined. Thus, convergence of $\nabla \ell(\tilde{\theta}(1)), \nabla \ell(\tilde{\theta}(2)), \ldots$ to zero cannot be established in a straightforward manner. Moreover, to show further that $\nabla \ell(\tilde{\theta}(1)), \nabla \ell(\tilde{\theta}(2)), \ldots$ converges to zero with probability one implies $\tilde{\theta}(1), \tilde{\theta}(2), \ldots$ converges to the set of critical points of ℓ with probability one typically requires that $\nabla \ell$ is continuous.

3 A Practical Convergence Analysis Recipe

In this section, a procedure for applying the new stochastic approximation theorem is provided. Section 5 provides a formal statement and proof of the theorem.

The assumption that a stochastic sequence $\tilde{\mathbf{x}}(1), \tilde{\mathbf{x}}(2), \ldots$ is bounded means that there exists some finite number K such that $|\tilde{\mathbf{x}}(t)| \leq K$ with probability one. Here, the random vector $\tilde{\mathbf{x}}(t)$ corresponds to an experiment that generates a training stimulus vector $\mathbf{x}(t)$. If the random vector $\tilde{\mathbf{x}}(t)$ is a discrete random vector restricted to take on a finite number of values (e.g., a d-dimensional binary random vector $\tilde{\mathbf{x}}(t) \in \{0, 1\}^d$), then this is a sufficient condition for the stochastic sequence to be bounded.

A sufficient condition for $c(\tilde{\mathbf{x}}, \boldsymbol{\theta})$ to be called a twice continuously differentiable random function is if c is a continuous function of $\tilde{\mathbf{x}}$ and the second

derivative of c, $\mathbf{H}$, is a continuous function on the q-dimensional parameter space Θ.

The conclusion of the convergence theorem states that the stochastic sequence of parameter estimates $\tilde{\boldsymbol{\theta}}(1), \tilde{\boldsymbol{\theta}}(2)$ either (1) is not confined to a closed, bounded, and convex region, Θ, of the parameter space with probability one, or (2) converges to the set of critical points in Θ with probability one. For example, if the stochastic sequence of parameter estimates $\tilde{\boldsymbol{\theta}}(1), \tilde{\boldsymbol{\theta}}(2)$ converges to a set of two critical points of ℓ such that it oscillates between these two points forever with probability one, then the stochastic sequence of parameter estimates $\tilde{\boldsymbol{\theta}}(1), \tilde{\boldsymbol{\theta}}(2)$ is said to converge to this set of two critical points with probability one:

- *Step 1: Identify the statistical environment.* A reactive statistical environment is modeled as a sequence of bounded, independent, and identically distributed d-dimensional random vectors $\tilde{\mathbf{x}}(1), \tilde{\mathbf{x}}(2), \ldots$ with common density $p_e(\cdot|\boldsymbol{\theta})$ where $\boldsymbol{\theta} \in \mathcal{R}^q$. The density p_e is not functionally dependent on $\boldsymbol{\theta}$ for passive statistical environments.
- *Step 2: Check ℓ is twice continuously differentiable with a lower bound.* Since $\{\tilde{\mathbf{x}}(t)\}$ is assumed bounded and it will be assumed that $\{\tilde{\boldsymbol{\theta}}(t)\}$ is a bounded stochastic sequence, this assumption is satisfied provided that c and p_e are twice continuously differentiable random functions and defined such that for all $\boldsymbol{\theta} \in \mathcal{R}^q$:

  $$\ell(\boldsymbol{\theta}) = \int c(\mathbf{x}, \boldsymbol{\theta}) p_e(\mathbf{x}|\boldsymbol{\theta}) d\nu(\mathbf{x}).$$

  That is, $\ell(\boldsymbol{\theta}) = E\{c(\tilde{\mathbf{x}}, \boldsymbol{\theta})\}$ where the expectation is taken with respect to $p_e(\mathbf{x}|\boldsymbol{\theta})$. It is also assumed that ℓ has a lower bound on $\mathcal{R}^q$.
- *Step 3: Define the region of convergence.* Let Θ be a closed, bounded, and convex subset of $\mathcal{R}^q$.
- *Step 4: Check the annealing schedule.* Define a sequence of step sizes $\gamma_1, \gamma_2, \ldots$ that satisfies equations 5.1 and 5.2. In the context of adaptive learning, γ_t corresponds to the adaptive learning algorithm's "learning rate." For example, the step-size schedule

  $$\gamma_t = \frac{\gamma_0 (1 + (t/\tau_1))}{1 + (t/\tau_2)^2},$$

  where $0 < \tau_1 < \tau_2$ and positive γ_0 generates a sequence $\gamma_1, \gamma_2, \ldots$ that satisfies special constraints on the step-size sequence specified by equations 5.1 and 5.2. This particular step-size schedule initially increases the step size and then eventually decreases it. The constant τ_1 should be chosen to be large enough that the learning algorithm observes a sufficiently rich sample of its statistical environment to support learning. The constant τ_2 should be the same order of magnitude as τ_1. So, for example, if the learning machine observes M distinct training stimuli with approximately equal probability and only one training stimulus is observed per iteration, then τ_1 might be chosen to

be $10M$ so that each training stimulus is observed approximately 10 times during both the search and the converge phases of the learning process.

- *Step 5: Identify the search direction function.* Let $\mathbf{d}_t : \mathcal{R}^d \times \mathcal{R}^q \to \mathcal{R}^q$ be a piecewise continuous function on $\mathcal{R}^d \times \mathcal{R}^q$ for each $t \in \mathbb{N}$. Rewrite the learning rule for updating parameter estimates using the formula

$$\tilde{\boldsymbol{\theta}}(t+1) = \tilde{\boldsymbol{\theta}}(t) + \gamma_t \tilde{\mathbf{d}}_t,$$

where the search direction random vector $\tilde{\mathbf{d}}_t = \mathbf{d}_t(\tilde{\mathbf{x}}(t), \tilde{\boldsymbol{\theta}}(t))$, and $\{\tilde{\mathbf{d}}_t\}$ is a bounded stochastic sequence. A sufficient condition for $\{\tilde{\mathbf{d}}_t\}$ to be a bounded stochastic sequence is that there exists a piecewise continuous function $\mathbf{d} : \mathcal{R}^d \times \mathcal{R}^q \to \mathcal{R}^q$ on a finite partition of $\mathcal{R}^d \times \mathcal{R}^q$ such that $\mathbf{d}_t = \mathbf{d}$ for all $t \in \mathbb{N}$ since $\{\tilde{\mathbf{x}}(t)\}$ and $\{\tilde{\boldsymbol{\theta}}(t)\}$ are bounded stochastic sequences by assumption.

- *Step 6: Show the average search direction is downward.* Assume there exists a series of functions $\bar{\mathbf{d}}_1, \bar{\mathbf{d}}_2, \ldots$ such that

$$\bar{\mathbf{d}}_t(\boldsymbol{\theta}) \equiv E\{\mathbf{d}_t(\tilde{\mathbf{x}}(t), \boldsymbol{\theta}) \,|\boldsymbol{\theta}\} = \int \mathbf{d}_t(\mathbf{x}, \boldsymbol{\theta})\, p_e(\mathbf{x}|\boldsymbol{\theta}) d\nu(\mathbf{x}).$$

Show that there exists a positive number K such that

$$\bar{\mathbf{d}}_t(\boldsymbol{\theta})^T \mathbf{g}(\boldsymbol{\theta}) \leq -K|\mathbf{g}(\boldsymbol{\theta})|^2. \tag{3.1}$$

For example, choosing

$$\begin{aligned}\mathbf{d}_t(\mathbf{x}, \boldsymbol{\theta}) &= -\left(\frac{1}{p_e(\mathbf{x}|\boldsymbol{\theta})}\right) \frac{d[c(\mathbf{x}, \boldsymbol{\theta}) p_e(\mathbf{x}|\boldsymbol{\theta})]}{d\boldsymbol{\theta}} \\ &= -\frac{dc(\mathbf{x}, \boldsymbol{\theta})}{d\boldsymbol{\theta}} - c(\mathbf{x}, \boldsymbol{\theta}) \frac{d \log p_e(\mathbf{x}|\boldsymbol{\theta})}{d\boldsymbol{\theta}}\end{aligned}$$

yields the standard stochastic gradient descent direction

$$\bar{\mathbf{d}}_t(\boldsymbol{\theta}) = \int \mathbf{d}_t(\mathbf{x}, \boldsymbol{\theta}) p_e(\mathbf{x}|\boldsymbol{\theta}) d\nu(\mathbf{x}) = -d\ell/d\boldsymbol{\theta}$$

so that $\bar{\mathbf{d}}_t(\boldsymbol{\theta})^T \mathbf{g}(\boldsymbol{\theta}) = -|\mathbf{g}(\boldsymbol{\theta})|^2$.

- *Step 7: Investigate asymptotic behavior.* Let $\mathcal{H}$ be the set of critical points in Θ. Conclude that with probability one either (1) the stochastic sequence does not remain in Θ for all $t > T$ for some positive integer T, or (2) $\tilde{\boldsymbol{\theta}}(t) \to \mathcal{H}$ as $t \to \infty$.

Consider the important special case where the Hessian of ℓ is positive definite on Θ even though ℓ is multimodal. The region Θ can contain no critical points, exactly one critical point, or multiple critical points. If Θ contains exactly one critical point in its interior, then that critical point is the unique global minimizer of ℓ on the interior of Θ. The region $\Theta \subseteq \mathcal{R}^q$ may also contain one or more critical points of ℓ on its boundary corresponding to saddle

points or local maximizers of ℓ on $\mathcal{R}^q$. For example, suppose that a smooth objective function ℓ has a strict local minimum at the point $\theta = 0$, a saddle point at $\theta = 5$, and a strict local maximum at the point $\theta = 10$. The function ℓ is positive definite on the set $\Theta_1 = [-3, -1]$ but no critical points exist in Θ_1. The function ℓ is positive definite on the set $\Theta_2 = [-3, +3]$ and has a unique strict local minimizer at $\theta = 0$. The function ℓ is positive definite on the set $\Theta_3 = [-3, 5]$ and has two critical points located at $\theta = 0$ (strict local minimizer) and $\theta = 5$ (critical point on boundary of Θ_3).

4 Adaptive Learning Algorithm Applications

In this section, we discuss several examples of adaptive learning algorithms that can be analyzed using the stochastic approximation theorem for reactive environments presented in section 5.

4.1 Adaptive Learning in Passive Statistical Environments. In this section, some adaptive learning strategies for passive statistical environments are discussed. In such environments, the objective function is defined as in equation 1.4. It should be noted, however, that these adaptive learning strategies are applicable for reactive learning statistical environments as well where the objective function is defined as in equation 1.5.

Assume the observations $\tilde{\mathbf{x}}(1), \tilde{\mathbf{x}}(2), \ldots$ are independent and identically distributed with common density p_e.

Let the notation

$$\tilde{\mathbf{g}}_k \equiv \left[\frac{dc(\tilde{\mathbf{x}}(k), \tilde{\boldsymbol{\theta}}(k))}{d\boldsymbol{\theta}}\right]^T, \tag{4.1}$$

where c is defined as in equation 1.4. A stochastic gradient descent (SGD) method (e.g., Bottou, 1991, 1998) corresponds to selecting the search direction,

$$\tilde{\mathbf{d}}_k = -\tilde{\mathbf{g}}_k. \tag{4.2}$$

The step-size γ_k is often chosen such that $k\gamma_k$ converges to a constant number as $k \to \infty$ in order to ensure that both equations 5.1 and 5.2 hold.

In practice, one often uses a minibatch stochastic approximation algorithm. Assume the kth minibatch $\tilde{\mathbf{X}}_k$ is constructed such that

$$\tilde{\mathbf{X}}_k \equiv [\tilde{\mathbf{x}}((k-1)m+1)), \ldots, \tilde{\mathbf{x}}(km)]$$

so that the stochastic sequence of minibatches $\{\tilde{\mathbf{X}}_k\}$ is also independent and identically distributed.

Then the adaptive learning update equation is given by

$$\tilde{\boldsymbol{\theta}}(k+1) = \tilde{\boldsymbol{\theta}}(k) + \gamma_k \tilde{\mathbf{d}}_k, \tag{4.3}$$

where the search direction

$$\tilde{\mathbf{d}}_k \equiv (1/m) \sum_{j=(k-1)m+1}^{km} \mathbf{d}_k(\tilde{\mathbf{x}}(j), \tilde{\boldsymbol{\theta}}(k)).$$

A modified-Newton algorithm (Ollivier, 2015; Roux et al., 2008; Schraudolph et al., 2007; Sunehag et al., 2009) is realized by the choice

$$\tilde{\mathbf{d}}_k = -\mathbf{M}_k \tilde{\mathbf{g}}_k, \tag{4.4}$$

where $\mathbf{M}_k$ is a positive-definite symmetric matrix chosen to approximate the inverse of the Hessian of ℓ evaluated at $\boldsymbol{\theta}(k)$ when $\boldsymbol{\theta}(k)$ is near a strict local minimizer of ℓ. Note that the Hessian of ℓ in the neighborhood of a strict local minimizer will have bounded strictly positive eigenvalues. This is an important observation since this ensures that for the case where equation 4.4 holds, the expected downhill condition in equation 3.1 also holds. That is,

$$\bar{\mathbf{d}}_k^T \mathbf{g} = -\mathbf{g}^T \mathbf{M}_k \mathbf{g} \leq -\lambda_{\min} |\mathbf{g}|^2,$$

where the positive number $\lambda_{\min}$ is the smallest eigenvalue of $\mathbf{M}_k$ for all $k \in \mathbb{N}$.

For nonconvex functions when $\boldsymbol{\theta}(k)$ is not near a strict local minimizer, the smallest eigenvalue of $\mathbf{M}_k$, λ_k, may be zero or negative. In such cases, one can take a Levenberg-Marquardt type step,

$$\tilde{\mathbf{d}}_k = -\mathbf{M}_k \tilde{\mathbf{g}}_k - \mu_k \tilde{\mathbf{g}}_k,$$

where μ_k is a positive number chosen to be greater than the positive number $\lambda_{\min}$ or one can simply choose a standard gradient descent step $\tilde{\mathbf{d}}_k = -\tilde{\mathbf{g}}_k$. This algorithm also ensures that $\bar{\mathbf{d}}_k^T \mathbf{g} \leq -\lambda_{\min} |\mathbf{g}|^2$.

The above methodology can also be used to implement different stochastic approximation variants of momentum, conjugate gradient, limited memory Broyden-Fletcher-Goldfarb-Shanno descent algorithms (Shraudolph et al., 2007; Jani et al., 2000; Paik et al., 2006), natural gradient descent methods (Schraudolph et al., 2007), and normalized gradient methods (Hazan et al., 2015).

Stochastic gradient descent with adaptive momentum (Pearlmutter, 1992; Roux et al., 2012; Sutskever et al., 2013; Zhang et al., 2015) is widely

used in the field of machine learning and is closely related to conjugate-gradient and other variable metric methods. Define the gradient descent search direction with momentum for $k = 1, 2, \ldots$ as

$$\tilde{\mathbf{d}}(k) = -\tilde{\mathbf{g}}_k + \tilde{\mu}\tilde{\mathbf{d}}(k-1), \tag{4.5}$$

where either (1) $\tilde{\mu}_k = 0$ yielding a gradient descent step or (2) $\tilde{\mu}_k \in (0, 1)$ yielding a momentum type step.

This type of algorithm can be realized within the proposed theoretical framework as follows. Let

$$\mathbf{M}_k = \mathbf{I} - \tilde{\mu}_k \tilde{\mathbf{d}}(k-1)\tilde{\mathbf{d}}(k-1)^T,$$

where

$$\tilde{\mu}_k = \frac{\mu}{\tilde{\mathbf{d}}(k-1)^T \tilde{\mathbf{g}}_k},$$

where μ is a positive number. This implies that

$$\tilde{\mathbf{d}}_k = -\mathbf{M}_k \tilde{\mathbf{g}}_k = -\tilde{\mathbf{g}}_k + \left(\frac{\tilde{\mu}\tilde{\mathbf{d}}(k-1)\tilde{\mathbf{d}}(k-1)^T}{\tilde{\mathbf{d}}(k-1)^T \tilde{\mathbf{g}}_k} \right) \tilde{\mathbf{g}}_k,$$

which can then be rewritten in the form of equation 4.5 provided that $\tilde{\mathbf{d}}(k-1)^T \tilde{\mathbf{g}}_k \neq 0$.

In practice, one would set $\tilde{\mu}_k = 0$, yielding a gradient descent step in situations where the magnitude of $\tilde{\mathbf{d}}(k-1)^T \tilde{\mathbf{g}}_k$ is less than some positive number ϵ.

Also note that $q - 1$ eigenvalues of $\mathbf{M}_t$ are equal to 1, and the remaining eigenvalue is $1 - \tilde{\mu}_k |\tilde{\mathbf{d}}(k-1)|^2$. Thus, to satisfy the conditions of the theorem so that the smallest eigenvalue of $\mathbf{M}_t$ is greater than a positive number $\lambda_{\min}$ and the largest eigenvalue of $\mathbf{M}_t$ is less than a positive number $\lambda_{\max}$, it is sufficient for

$$\lambda_{\min} < 1 - \tilde{\mu}_k |\tilde{\mathbf{d}}(k-1)|^2 < \lambda_{\max}$$

or, equivalently, for the case where $\lambda_{\max} = 1$,

$$0 < \tilde{\mu}_k < \frac{1 - \lambda_{\min}}{|\tilde{\mathbf{d}}(k-1)|^2}. \tag{4.6}$$

In practice, one could check at each step of the algorithm if condition (4.6) is satisfied. If condition (4.6) is not satisfied, one could set $\tilde{\mu}_k = 0$ to realize a

gradient descent step. This would ensure that the sequence of real symmetric matrices $\mathbf{M}_1, \mathbf{M}_2, \ldots$ is both positive definite and uniformly bounded such that the eigenvalues of matrix $\mathbf{M}_t$, $\{\lambda_t(1), \ldots, \lambda_t(q)\}$, for any sequence $\mathbf{M}_1, \mathbf{M}_2, \ldots$ satisfy the relation

$$0 < \lambda_{\min} \leq \lambda_t(k) \leq \lambda_{\max}$$

for all $t \in \mathbb{N}$ and for all $k \in \{1, \ldots, q\}$.

A random block coordinate descent algorithm (Razaviyayn, Hong, Luo, & Pang, 2014) can be realized within this proposed framework as well. Let $\odot$ denote the Hadamard product (element-by-element vector multiplication) operator. Let the set of q-dimensional binary vectors be denoted by $B \equiv \{0, 1\}^q$. Let $\mathbf{m}_t \in B$ be a q-dimensional binary vector whose jth element is a one if the jth element of the q-dimensional random vector $\boldsymbol{\theta}(k)$ is updated with information about training pattern $\mathbf{s}(t)$ at learning trial t.

In particular, assume that at learning trial t, the ordered pair

$$(\mathbf{s}_t, \mathbf{m}_t) \in \mathcal{R}^d \times B$$

is a realization of the mixed random vector

$$\tilde{\mathbf{x}}_t \equiv (\tilde{\mathbf{s}}_t, \tilde{\mathbf{m}}_t),$$

whose Radon-Nikodým density is p_e with respect to sigma-finite measure ν. It is assumed that $\tilde{\mathbf{s}}_t$ and $\tilde{\mathbf{m}}_t$ are independent so that $p_e(\mathbf{s}, \mathbf{m}) = p_s(\mathbf{s})p_m(\mathbf{m})$.

Let the objective function for learning $\ell : \mathcal{R}^q \rightarrow \mathcal{R}^q$ be defined such that

$$\ell(\boldsymbol{\theta}) = \int c((\mathbf{s}, \mathbf{m}), \boldsymbol{\theta}) p_s(\mathbf{s}) p_m(\mathbf{m}) d\nu(\mathbf{s}, \mathbf{m}).$$

The search direction for random block coordinate gradient descent is defined as

$$\tilde{\mathbf{d}}_t = \mathbf{f}((\tilde{\mathbf{s}}_t, \tilde{\mathbf{m}}_t), \tilde{\boldsymbol{\theta}}(t)),$$

where

$$\mathbf{f}((\tilde{\mathbf{s}}_t, \tilde{\mathbf{m}}_t), \tilde{\boldsymbol{\theta}}(t)) = -\tilde{\mathbf{m}}_t \odot \tilde{\mathbf{g}}_t, \tag{4.7}$$

resulting in the random block coordinate descent adaptive learning rule:

$$\tilde{\boldsymbol{\theta}}(t+1) = \tilde{\boldsymbol{\theta}}(t) - \gamma_t \tilde{\mathbf{m}}_t \odot \tilde{\mathbf{g}}_t.$$

Let the expected frequency that a state vector element is updated be defined as

$$\bar{\mathbf{m}} = \sum_{\mathbf{m} \in B} \mathbf{m}\ p_m(\mathbf{m}).$$

The expected value of $\bar{\mathbf{d}}^T \tilde{\mathbf{g}}_t$ is given by

$$\bar{\mathbf{d}}^T \mathbf{g} = -\left(\bar{\mathbf{m}} \odot \mathbf{g}\right)^T \mathbf{g} \leq -m_{\min} |\mathbf{g}|^2,$$

where $m_{\min}$ is the smallest number in $\bar{\mathbf{m}}$. Thus, equation 5.4 holds provided that the expected frequency of times that each element of the state vector is updated is strictly positive (i.e., $m_{\min} > 0$).

4.2 Normalization Constants and Contrastive Divergence. Maximum likelihood estimation is a method for computing the parameter estimates that maximize the likelihood of the observed data or, equivalently, minimize the cross-entropy between the researcher's model and the empirical distribution of the observed data. For example, suppose that the observed data are a collection of n d-dimensional vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, which are presumed to be a particular realization of a sequence of independent and identically distributed random vectors with common density $p_e : \mathcal{R}^d \to (0, \infty)$ with respect to sigma-finite measure ν. Then the method of maximum likelihood estimation corresponds to finding the parameter vector $\hat{\boldsymbol{\theta}}_n$ that is a global minimizer of

$$\ell_n(\boldsymbol{\theta}) \equiv -(1/n) \sum_{i=1}^{n} \log p(\mathbf{x}_i | \boldsymbol{\theta}) \tag{4.8}$$

on Θ. In addition, as $n \to \infty$, $\hat{\boldsymbol{\theta}}_n \to \boldsymbol{\theta}^*$ with probability one where $\boldsymbol{\theta}^*$ is a particular global minimizer of

$$\ell(\boldsymbol{\theta}) = -\int p_e(\mathbf{x}) \log p(\mathbf{x}|\boldsymbol{\theta}) d\nu(\mathbf{x}) \tag{4.9}$$

under appropriate regularity conditions.

Let $V : \mathcal{R}^d \times \mathcal{R}^q \to \mathcal{R}$. Let Θ be a closed and bounded subset of $\mathcal{R}^q$. Assume for each $\boldsymbol{\theta} \in \Theta$ that the probability density of $\tilde{\mathbf{x}}$ is a Gibbs density $p(\cdot|\boldsymbol{\theta}) : \mathcal{R}^d \to (0, \infty)$ defined such that

$$p(\mathbf{x}|\boldsymbol{\theta}) = [Z(\boldsymbol{\theta})]^{-1} \exp(-V(\mathbf{x}; \boldsymbol{\theta})), \tag{4.10}$$

where the normalization constant $Z(\boldsymbol{\theta})$ is defined as

$$Z(\boldsymbol{\theta}) = \int \exp(-V(\mathbf{y};\boldsymbol{\theta}))d\nu(\mathbf{y}). \tag{4.11}$$

The derivative of ℓ_n in equation 4.8 is given by the formula

$$\frac{d\ell_n}{d\boldsymbol{\theta}} = (1/n)\sum_{i=1}^{n}\frac{d\ell_{n,i}}{d\boldsymbol{\theta}}, \tag{4.12}$$

where

$$\frac{d\ell_{n,i}}{d\boldsymbol{\theta}} = \frac{dV(\mathbf{x}_i,\boldsymbol{\theta})}{d\boldsymbol{\theta}} - \int \frac{dV(\mathbf{y};\boldsymbol{\theta})}{d\boldsymbol{\theta}} p(\mathbf{y}|\boldsymbol{\theta})d\nu(\mathbf{y}). \tag{4.13}$$

Equation 4.12 cannot, however, be immediately used to derive a stochastic gradient descent algorithm that minimizes ℓ for the following reasons. The first term on the right-hand side of equation 4.13 is usually relatively easy to evaluate. But the second term on the right-hand side of equation 4.13 is usually very difficult to evaluate because it involves a computationally intractable multidimensional integration.

Let $\tilde{\mathbf{y}}^1, \ldots, \tilde{\mathbf{y}}^m$ be a sequence of m possibly correlated distributed random vectors with a common mean whose joint density is $p(\mathbf{y}^1, \ldots, \mathbf{y}^m|\boldsymbol{\theta})$ for a given $\boldsymbol{\theta}$. To obtain a computationally practical method of evaluating the second term on the right-hand side of equation 4.13, note that the expected value of

$$(1/m)\sum_{j=1}^{m}\frac{dV(\tilde{\mathbf{y}}^j;\boldsymbol{\theta})}{d\boldsymbol{\theta}} \tag{4.14}$$

is

$$\int \frac{dV(\mathbf{y};\boldsymbol{\theta})}{d\boldsymbol{\theta}} p(\mathbf{y}|\boldsymbol{\theta})d\nu(\mathbf{y}), \tag{4.15}$$

which corresponds to the second term on the right-hand side of equation 4.13.

Substitute the Monte Carlo approximation in equation 4.14 for the multidimensional integral in equation 4.13 and then using the resulting approximate derivative as a stochastic search direction for a stochastic approximation algorithm defined by

$$\tilde{\boldsymbol{\theta}}(k+1) = \tilde{\boldsymbol{\theta}}(k) - \gamma_k\frac{dV(\tilde{\mathbf{x}}(k),\tilde{\boldsymbol{\theta}}(k))}{d\boldsymbol{\theta}} + (\gamma_k/m)\sum_{j=1}^{m}\frac{dV(\tilde{\mathbf{y}}^j;\tilde{\boldsymbol{\theta}}(k))}{d\boldsymbol{\theta}}, \tag{4.16}$$

where the minibatch $\tilde{\mathbf{y}}^1, \ldots, \tilde{\mathbf{y}}^m$ is a collection of m possibly highly correlated observations with joint density $p(\mathbf{y}^1, \ldots, \mathbf{y}^m|\boldsymbol{\theta}(k))$ for the kth iteration of equation 4.16. It is assumed that given $\tilde{\boldsymbol{\theta}}(k)$ the mini-batches are independent and identically distributed with common density $p(\mathbf{y}^1, \ldots, \mathbf{y}^m|\boldsymbol{\theta}(k))$. Equation 4.16 is an example of a contrastive divergence type of learning algorithm, which can be interpreted as a stochastic approximation algorithm. The minibatch size m can be a fixed integer (e.g., $m = 3$ or $m = 100$), or m can be varied (e.g., initially m is chosen to be small and then gradually increased to some finite positive integer during the learning process).

Note that the statistical environment used to generate the data for the stochastic approximation algorithm in equation 4.16 is not a passive statistical environment since the parameters of the learning machine are updated at learning trial k not only by the observation $\tilde{\mathbf{x}}(k)$ but also by the observations $\tilde{\mathbf{y}}^1, \ldots, \tilde{\mathbf{y}}^m$ whose joint distribution is functionally dependent on the current parameter estimates $\boldsymbol{\theta}(k)$. Thus, contrastive-divergence algorithms of this type can be analyzed approximately using the theorem presented in section 1.

4.3 Missing Data, Hidden Variables, and the EM Algorithm. In this section, the problems of hidden variables and missing data are considered. The presence of hidden variables is not only a characteristic feature of latent variable models and deep learning architectures but can be considered equivalent to the presence of data, which is always missing.

Assume that the data generating process generates a sequence of independent and identically distributed random vectors,

$$(\tilde{\mathbf{x}}_1, \tilde{\mathbf{m}}_1), (\tilde{\mathbf{x}}_2, \tilde{\mathbf{m}}_2), \ldots$$

where the d-dimensional random vector $\tilde{\mathbf{x}}_k$ is called a *complete-data* random vector and $\tilde{\mathbf{m}}_k$ is a d-dimensional *missing data indicator* binary random vector taking on values in $\{0, 1\}^d$ for all $k \in \mathbb{N}$. The jth element of $\tilde{\mathbf{m}}_k$ takes on the value of one if and only if the jth element of $\tilde{\mathbf{x}}_k$ is observable.

For convenience, the d-dimensional random vector $\tilde{\mathbf{x}}_k$ is partitioned such that $\tilde{\mathbf{x}}_k = [\tilde{\mathbf{v}}_k, \tilde{\mathbf{h}}_k]$ where $\tilde{\mathbf{v}}_k$ is the observable component of $\tilde{\mathbf{x}}_k$ and $\tilde{\mathbf{h}}_k$ is the unobservable component whose probability distribution is functionally dependent only on a realization of $\tilde{\mathbf{v}}_k$. The elements of $\tilde{\mathbf{v}}_k$ correspond to the visible random variables, while the elements of $\tilde{\mathbf{h}}_k$ correspond to the hidden random variables or the missing data. Note that the dimensionalities of $\tilde{\mathbf{v}}_k$ and $\tilde{\mathbf{h}}_k$ will typically vary as a function of the positive integer index variable t.

The missing data negative log-likelihood analogous to the complete data negative log likelihood in equation 4.8 is then defined by

$$\ell_n(\boldsymbol{\theta}) = -(1/n)\sum_{i=1}^{n} \log p(\mathbf{v}_i|\boldsymbol{\theta}), \tag{4.17}$$

which can be rewritten in terms of the joint density $p(\mathbf{v}_i, \mathbf{h}_i|\boldsymbol{\theta})$ as

$$\ell_n(\boldsymbol{\theta}) = -(1/n)\sum_{i=1}^{n} \log\left[\int p(\mathbf{v}_i, \mathbf{h}_i|\boldsymbol{\theta}) d\nu(\mathbf{h}_i)\right]. \tag{4.18}$$

Now take the derivative of equation 4.17 under the assumption that the interchange of derivative and integral operators is permissible to obtain

$$\frac{d\ell_n}{d\boldsymbol{\theta}} = (1/n)\sum_{i=1}^{n} \frac{d\ell_{i,n}}{d\boldsymbol{\theta}}, \tag{4.19}$$

where

$$\frac{d\ell_{i,n}}{d\boldsymbol{\theta}} = -\int \frac{1}{p(\mathbf{v}_i|\boldsymbol{\theta})} \frac{dp(\mathbf{v}_i, \mathbf{h}_i|\boldsymbol{\theta})}{d\boldsymbol{\theta}} d\nu(\mathbf{h}_i). \tag{4.20}$$

The derivative in the integrand of equation 4.20 is obtained using the identity (see Louis, 1982; McLachlan & Krishnan, 1996)

$$\frac{dp(\mathbf{v}_i, \mathbf{h}_i|\boldsymbol{\theta})}{d\boldsymbol{\theta}} = \frac{d\log p(\mathbf{v}_i, \mathbf{h}_i|\boldsymbol{\theta})}{d\boldsymbol{\theta}} p(\mathbf{v}_i, \mathbf{h}_i|\boldsymbol{\theta}). \tag{4.21}$$

Substitution of equation 4.21 into 4.20 gives

$$\frac{d\ell_{i,n}}{d\boldsymbol{\theta}} = -\int \frac{d\log p(\mathbf{v}_i, \mathbf{h}_i|\boldsymbol{\theta})}{d\boldsymbol{\theta}} p(\mathbf{h}_i|\mathbf{v}_i, \boldsymbol{\theta}), d\nu(\mathbf{h}_i),$$

which is then approximated using a Monte Carlo approximation using the formula

$$\frac{d\ell_{i,n}}{d\boldsymbol{\theta}} \approx -(1/m)\sum_{j=1}^{m} \frac{d\log\left[p(\mathbf{v}_i, \mathbf{h}^j|\boldsymbol{\theta})\right]}{d\boldsymbol{\theta}}, \tag{4.22}$$

where the stochastic imputation $\mathbf{h}^j$ is a realization of $\tilde{\mathbf{h}}^j$ whose distribution is specified by the conditional density $p(\mathbf{h}|\mathbf{v}, \boldsymbol{\theta})$ for a given realization $\mathbf{v}$ and parameter vector $\boldsymbol{\theta}$.

The final stochastic descent expectation-maximization algorithm is then constructed by defining the stochastic search direction as a negative one

multiplied by the derivative in equation 4.20 and then replacing the integral in it with the Monte Carlo approximation in equation 4.22 to yield the stochastic gradient descent algorithm:

$$\tilde{\boldsymbol{\theta}}(k+1) = \tilde{\boldsymbol{\theta}}(k) - (\gamma_k/m) \sum_{j=1}^{m} \frac{d \log \left[p(\tilde{\mathbf{v}}_i, \tilde{\mathbf{h}}^j | \boldsymbol{\theta}) \right]}{d\boldsymbol{\theta}},$$

where the minibatch $\tilde{\mathbf{h}}^1, \ldots, \tilde{\mathbf{h}}^m$ at the kth learning trial is generated by first sampling a realization $\mathbf{v}_i$ from the environment and then sampling m times from $p(\mathbf{h}|\mathbf{v}_i, \boldsymbol{\theta}(k))$ using the sampled value $\mathbf{v}_i$ and the current parameter estimates $\boldsymbol{\theta}(k)$ at the kth learning trial. Thus, the new stochastic approximation theorem provides a method for analyzing the asymptotic behavior of the stochastic descent expectation-maximization algorithm.

Note that m can be chosen equal to 1 or any positive integer. In the case where $m = \infty$, the resulting algorithm approximates the deterministic generalized expectation-maximization (GEM) algorithm (see McLachlan & Krishnan, 1996, for a formal definition of a GEM algorithm) in which the learning machine uses its current probabilistic model to compute the expected downhill search direction, takes a downhill step, updates its current probabilistic model, and then repeats this process in an iterative manner.

4.4 Policy Gradient Reinforcement Learning. In this section, the stochastic approximation theorem developed here is applied to the problem of investigating the convergence of a class of reinforcement learning algorithms called *policy gradient reinforcement learning machines* (Williams, 1992; Sutton & Barto, 1998; Sugiyama, 2015). Suppose that a learning machine experiences a collection of episodes. The episodes $\tilde{\mathbf{u}}(0), \tilde{\mathbf{u}}(1), \ldots$ are assumed to be independent and identically distributed. In addition, the kth episode $\mathbf{u}(k)$ is defined such that $\mathbf{u}(k) \equiv [\mathbf{s}_o(k), \mathbf{s}_F(k)]$ where $\mathbf{s}_o(k)$ is called the *initial state of episode* $\mathbf{u}(k)$ and $\mathbf{s}_F(k)$ is called the *final state of episode* $\mathbf{u}(k)$. The probability density of $\tilde{\mathbf{u}}_k$ when the learning machine is embedded within a passive statistical environment is specified by the density $p_e(\mathbf{u}) = p_e(\mathbf{s}_o, \mathbf{s}_F)$ where $p_e(\mathbf{u})$ specifies the likelihood that $\mathbf{u}$ is observed by the learning machine in its statistical environment.

On the other hand, for a reactive learning environment, the probability that the learning machine selects action $\mathbf{a}_j$ given the current state of the environment $\mathbf{s}_o$ and the learning machine's current state of knowledge $\boldsymbol{\theta}$ is expressed by the conditional probability mass function $p(\mathbf{a}_j|\mathbf{s}_o, \boldsymbol{\theta})$, $j = 1, \ldots, J$. The statistical environment of the learning machine is characterized by the probability density $p_e(\mathbf{s}_o)$, specifying the likelihood of a given initial state of an episode and the conditional density $p_e(\mathbf{s}_F|\mathbf{a}_j, \mathbf{s}_o)$, which specifies the likelihood of a final state of an episode $\mathbf{s}_F$ given the learning machine's action $\mathbf{a}_j$ and the initial state of the episode $\mathbf{s}_o$.

Thus, the probability distribution of an episode $\tilde{\mathbf{u}}(k)$ is specified by the density

$$p_e(\mathbf{u}|\boldsymbol{\theta}) = p_e(\mathbf{s}_o)p(\mathbf{s}_F|\mathbf{s}_o, \boldsymbol{\theta}),$$

where

$$p(\mathbf{s}_F|\mathbf{s}_o, \boldsymbol{\theta}) \equiv \sum_{j=1}^{J} p_e(\mathbf{s}_F|\mathbf{a}_j, \mathbf{s}_o)p(\mathbf{a}_j|\mathbf{s}_o, \boldsymbol{\theta}).$$

Let $c(\mathbf{u}; \boldsymbol{\theta})$ specify the cost incurred by the learning machine when episode $\mathbf{u}$ is encountered in its environment for a particular state of knowledge $\boldsymbol{\theta}$. Notice that the cost $c(\mathbf{u}; \boldsymbol{\theta})$ is functionally dependent on $\boldsymbol{\theta}$ as well as $\mathbf{u}$, allowing for the possibility of a learning machine with an "adaptive critic" (Sutton & Barton, 1998). One possible goal of an adaptive learning machine in a reactive statistical environment is to minimize the objective function ℓ defined by the formula

$$\ell(\boldsymbol{\theta}) = \int c(\mathbf{u}, \boldsymbol{\theta})p_e(\mathbf{u}|\boldsymbol{\theta})d\nu(\mathbf{u}), \tag{4.23}$$

where $p_e(\cdot|\boldsymbol{\theta})$ is a density for each $\boldsymbol{\theta} \in \mathcal{R}^q$.

Now take the derivative of equation 4.23, interchange the integral and derivative operators, and use a Monte Carlo approximation for the integral in that equation similar to the approximations in equations 4.14 and 4.22. In order to obtain the derivative of equation 4.23 in an appropriate form, the identity

$$\frac{dp_e(\mathbf{u}|\boldsymbol{\theta})}{d\boldsymbol{\theta}} = p_e(\mathbf{u}|\boldsymbol{\theta})\frac{d\log p_e(\mathbf{u}|\boldsymbol{\theta})}{d\boldsymbol{\theta}}$$

is used (see Louis, 1982; McLachlan & Krishnan, 1996).

The resulting derivative can then be used to construct the stochastic gradient descent algorithm, which works by updating the parameters of the learning machine after each episode using the formula

$$\tilde{\boldsymbol{\theta}}(k+1) = \tilde{\boldsymbol{\theta}}(k) - \gamma_k \frac{dc(\tilde{\mathbf{u}}(k), \tilde{\boldsymbol{\theta}})}{d\boldsymbol{\theta}} - \gamma_k c(\mathbf{u}, \tilde{\boldsymbol{\theta}}(k))\frac{d\log p_e(\tilde{\mathbf{u}}(k)|\tilde{\boldsymbol{\theta}}(k))}{d\boldsymbol{\theta}}. \tag{4.24}$$

Note that the probability distribution of $\tilde{\mathbf{u}}(k)$ at learning trial k is specified by the conditional probability density $p_e(\mathbf{u}(k)|\boldsymbol{\theta}(k))$.

5 Formal Convergence Analysis of Learning

In this section, the proof of the stochastic approximation theorem is provided, which minimizes the reactive environment risk function in equation 1.5 as well as the passive environment risk function in equation 1.4.

Although the specific theorem and proof presented here are novel, the obtained results and method of proof are very similar to many existing results in the literature. In particular, the statement and proof of the theorem follow a combination of arguments by Blum (1954), the appendix of Benveniste et al. (1990), and Sunehag et al. (2009) using the the well-known Robbins-Siegmund lemma (Robbins & Siegmund, 1971; see Benveniste et al., 1990, appendix to part 2, or Douc, Moulines, & Stoffer, 2014, lemma C2, for relevant reviews).

The results presented here are similar to those obtained by Andrieu et al. (2005, theorem 2.3), Benveniste et al. (1990, appendix to part 2, pp. 344–347), Bertsekas & Tsitsiklis (1996, proposition 4.1, p. 141), Douc et al. (2014, theorem C.7), Kushner (1981, theorem 1), Kushner & Yin (1997, theorem 4.1), Mohri et al (2012, theorems 14.7 and 14.8), White (1989a, 1989b, theorem 3.1).

The terminology that a function $\mathbf{f} : \mathcal{R}^d \times \mathcal{R}^q \to \mathcal{R}^q$ is *bounded* means that for all $(\mathbf{x}, \boldsymbol{\theta}) \in \mathcal{R}^d \times \mathcal{R}^q$, there exists a finite number K such that $|\mathbf{f}| \leq K$. The terminology that a stochastic sequence $\tilde{\mathbf{x}}(0), \tilde{\mathbf{x}}(1), \ldots$ is *bounded* means that there exists a finite number K such that for all $t \in \mathbb{N}$: $|\tilde{\mathbf{x}}(t)| \leq K$ with probability one where $\mathbb{N} \equiv \{0, 1, 2, \ldots\}$.

Let $\mathcal{D} \equiv \{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$ be a finite partition of $\mathcal{R}^d$. Let $\phi_1, \ldots, \phi_M$ be a finite set of functions defined such that $\phi_k(\mathbf{x}) = 1$ if $\mathbf{x} \in \mathcal{D}_k$ and $\phi_k(\mathbf{x}) = 0$ if $\mathbf{x} \notin \mathcal{D}_k$. Let $f_k : \mathcal{R}^d \to \mathcal{R}$ be a continuous function, $k = 1, \ldots, M$. The function $f : \mathcal{R}^d \to \mathcal{R}$ defined such that for all $\mathbf{x} \in \mathcal{R}^d$,

$$f(\mathbf{x}) = \sum_{k=1}^{M} f_k(\mathbf{x})\phi_k(\mathbf{x}),$$

is called a *piecewise continuous function on the finite partition* $\mathcal{D}$.

Let Θ be a convex, closed, and bounded subset of $\mathcal{R}^q$. Let $\ell : \mathcal{R}^q \to \mathcal{R}$ be a twice continuously differentiable function.

Let the gradient of ℓ be denoted as $\mathbf{g} \equiv (\nabla \ell)^T$. Let the Hessian of ℓ be denoted as $\mathbf{H} \equiv \nabla^2 \ell$.

Theorem 1. Almost Supermartingale Lemma (Special Case). *Let $\tilde{x}_1, \tilde{x}_2, \ldots$ be a stochastic sequence. Let $\tilde{r}_1, \tilde{r}_2, \ldots$ be a stochastic sequence of nonnegative random variables such that as $T \to \infty$, $\sum_{t=1}^{T} \tilde{r}_t$ converges to a finite number with probability one. Let $\phi : \mathcal{R}^d \to [0, \infty)$ and $V : \mathcal{R}^d \to [0, \infty)$ be nonnegative piecewise continuous functions on finite partitions of $\mathcal{R}^d$. Let $\tilde{q}_t \equiv \phi(\tilde{\mathbf{x}}_t)$ and $\tilde{v}_t \equiv V(\tilde{\mathbf{x}}_t)$ for all $t \in \mathbb{N}$. Assume, in addition, that for all $t \in \mathbb{N}$,*

$$E\{V(\tilde{\boldsymbol{x}}_{t+1})|\boldsymbol{x}_t\} \leq V(\boldsymbol{x}_t) - \phi(\boldsymbol{x}_t) + \tilde{r}_t.$$

Then $\tilde{v}_t$ converges to a random variable with probability one as $t \to \infty$, and there exists a finite number K such that as $T \to \infty$: $\sum_{t=1}^{T} \tilde{q}_t < K$ with probability one.

See Robbins and Siegmund (1971; also see Benveniste et al., 1990, p. 344, or Douc et al., 2014, lemma C2) for the statement and proof of the almost supermartingale lemma.

Theorem 2 (stochastic approximation theorem). *Let Θ be a closed, bounded, and convex subset of $\mathcal{R}^q$. Let $\ell : \mathcal{R}^q \to \mathcal{R}$ be a twice continuously differentiable function with a finite lower bound. Let $\boldsymbol{g} \equiv (\nabla \ell)^T$. Let $\boldsymbol{H} \equiv \nabla^2 \ell$.*

- *Assume $\tilde{\boldsymbol{x}}_{\boldsymbol{\theta}}$ has Radon-Nikodým density $p_e(\cdot|\boldsymbol{\theta}) : \mathcal{R}^d \to [0, \infty)$ with respect to a sigma-finite measure ν for each $\boldsymbol{\theta} \in \Theta$.*
- *Assume a positive number x_{max} exists such that for all $\boldsymbol{\theta} \in \Theta$, the random vector $\tilde{\boldsymbol{x}}_{\boldsymbol{\theta}}$ with density $p_e(\cdot|\boldsymbol{\theta})$ satisfies $|\tilde{\boldsymbol{x}}_{\boldsymbol{\theta}}| < x_{max}$ with probability one.*
- *Let $\gamma_0, \gamma_1, \gamma_2, \ldots$ be a sequence of positive real numbers such that*

$$\sum_{t=0}^{\infty} \gamma_t^2 < \infty \tag{5.1}$$

and

$$\sum_{t=0}^{\infty} \gamma_t = \infty. \tag{5.2}$$

- *Let $\boldsymbol{d}_t : \mathcal{R}^d \times \mathcal{R}^q \to \mathcal{R}^q$ be a piecewise continuous function on a finite partition of $\mathcal{R}^d \times \mathcal{R}^q$ for all $t \in \mathbb{N}$. When it exists, let*

$$\bar{\boldsymbol{d}}_t(\boldsymbol{\theta}) = \int \boldsymbol{d}_t(\boldsymbol{x}, \boldsymbol{\theta}) p_e(\boldsymbol{x}|\boldsymbol{\theta}) d\nu(\boldsymbol{x}).$$

- *Let $\tilde{\boldsymbol{\theta}}(0)$ be a q-dimensional random vector. Let $\tilde{\boldsymbol{\theta}}(1), \tilde{\boldsymbol{\theta}}(2), \ldots$ be a sequence of q-dimensional random vectors defined such that for $t = 0, 1, 2, \ldots$,*

$$\tilde{\boldsymbol{\theta}}(t+1) = \tilde{\boldsymbol{\theta}}(t) + \gamma_t \tilde{\boldsymbol{d}}_t, \tag{5.3}$$

where $\tilde{\boldsymbol{d}}_t \equiv \boldsymbol{d}_t(\tilde{\boldsymbol{x}}_{\boldsymbol{\theta}}(t), \tilde{\boldsymbol{\theta}}(t))$ such that $|\tilde{\boldsymbol{d}}_t|$ is less than some finite number for $t = 0, 1, 2, \ldots$, and the distribution of $\tilde{\boldsymbol{x}}_{\boldsymbol{\theta}}(t)$ is specified by the conditional density $p_e(\cdot|\tilde{\boldsymbol{\theta}}(t))$.
- *Assume there exists a positive number K such that for all $\boldsymbol{\theta} \in \Theta$,*

$$\bar{\boldsymbol{d}}_t(\boldsymbol{\theta})^T \boldsymbol{g}(\boldsymbol{\theta}) \leq -K|\boldsymbol{g}(\boldsymbol{\theta})|^2. \tag{5.4}$$

If there exists a positive integer T such that $\tilde{\boldsymbol{\theta}}(t) \in \Theta$ for all $t \geq T$ with probability one, then $\tilde{\boldsymbol{\theta}}(1), \tilde{\boldsymbol{\theta}}(2), \ldots$ converges with probability one to the set of critical points of ℓ contained in Θ.

Proof. Let $\tilde{\ell}_t \equiv \ell(\tilde{\boldsymbol{\theta}}(t))$ with realization $\ell_t \equiv \ell(\boldsymbol{\theta}(t))$. Let $\tilde{\mathbf{g}}_t \equiv \mathbf{g}(\tilde{\boldsymbol{\theta}}(t))$ with realization $\mathbf{g}_t \equiv \mathbf{g}(\boldsymbol{\theta}(t))$. Let $\tilde{\mathbf{H}}_t \equiv \mathbf{H}(\tilde{\boldsymbol{\theta}}(t))$ with realization $\mathbf{H}_t \equiv \mathbf{H}(\boldsymbol{\theta}(t))$.

Step 1: Expand ℓ using a second-order mean value expansion. Expand ℓ about $\tilde{\boldsymbol{\theta}}(t)$ and evaluate at $\tilde{\boldsymbol{\theta}}(t+1)$ using the mean value theorem to obtain

$$\tilde{\ell}_{t+1} = \tilde{\ell}_t + \tilde{\mathbf{g}}_t^T \left(\tilde{\boldsymbol{\theta}}(t+1) - \tilde{\boldsymbol{\theta}}(t)\right) + \gamma_t^2 \tilde{R}_t \tag{5.5}$$

with

$$\tilde{R}_t \equiv (1/2)\tilde{\mathbf{d}}_t^T \mathbf{H}(\tilde{\boldsymbol{\zeta}}_t)\tilde{\mathbf{d}}_t, \tag{5.6}$$

where the random variable $\tilde{\boldsymbol{\zeta}}_t$ can be defined as a point on the chord connecting $\tilde{\boldsymbol{\theta}}(t)$ and $\tilde{\boldsymbol{\theta}}(t+1)$. Substituting the relation

$$\gamma_t \tilde{\mathbf{d}}_t = \tilde{\boldsymbol{\theta}}(t+1) - \tilde{\boldsymbol{\theta}}(t)$$

into equation 5.5 gives

$$\tilde{\ell}_{t+1} = \tilde{\ell}_t + \gamma_t \tilde{\mathbf{g}}_t^T \tilde{\mathbf{d}}_t + \gamma_t^2 \tilde{R}_t. \tag{5.7}$$

Step 2: Identify conditions required for the remainder term of the expansion to be bounded. Since, by assumption, $\{\tilde{\boldsymbol{\theta}}(t)\}$ is a bounded stochastic sequence and $\mathbf{H}$ is continuous, this implies that the stochastic sequence $\{\mathbf{H}(\tilde{\boldsymbol{\zeta}}_t)\}$ is bounded. In addition, by assumption, $\{\tilde{\mathbf{d}}_t\}$ is a bounded stochastic sequence. This implies there exists a number $R_{\max}$ such that for all $t = 0, 1, 2, \ldots,$

$$|\tilde{R}_t| < R_{\max}, \tag{5.8}$$

with probability one.

Step 3: Show the expected value of objective function decreases. Taking the conditional expectation of both sides of equation 5.7 with respect to the conditional density p_e and evaluating at $\boldsymbol{\theta}(t)$ and γ_t yields

$$E\left\{\tilde{\ell}_{t+1}|\boldsymbol{\theta}(t)\right\} = \ell_t + \gamma_t \mathbf{g}_t^T \bar{\mathbf{d}}_t + \gamma_t^2 E\{\tilde{R}_t|\boldsymbol{\theta}(t)\}. \tag{5.9}$$

Substituting the assumption $\bar{\mathbf{d}}_t(\boldsymbol{\theta})^T \mathbf{g}(\boldsymbol{\theta}) \leq -K|\mathbf{g}(\boldsymbol{\theta})|^2$ and the conclusion of step 2 that $|\tilde{R}_t| < R_{\max}$ with probability one into equation 5.7 gives

$$E\left\{\tilde{\ell}_{t+1}|\boldsymbol{\theta}(t)\right\} \leq \ell_t - \gamma_t K|\mathbf{g}_t|^2 + \gamma_t^2 R_{\max}. \tag{5.10}$$

Step 4: Show a subsequence of $\{|\tilde{\mathbf{g}}_t|^2\}$ converges to zero wp1. Since ℓ has a lower bound, K is a finite positive number, and equation 5.1 holds by assumption, then the almost supermartingale lemma can be applied

to equation 5.10 on the set where $\{\tilde{\boldsymbol{\theta}}(t)\}$ and $\{\tilde{\mathbf{d}}_t\}$ are bounded with probability one to obtain the conclusion that

$$\sum_{t=0}^{\infty} \gamma_t \left|\tilde{\mathbf{g}}_t\right|^2 < \infty \tag{5.11}$$

with probability one.

For some positive integer T, let

$$\tilde{a}_T^* \equiv \inf\left\{\left|\tilde{\mathbf{g}}_T\right|^2, \left|\tilde{\mathbf{g}}_{T+1}\right|^2, \ldots\right\}.$$

The sequence $\tilde{a}_T^*, \tilde{a}_{T+1}^*, \ldots$ is nonincreasing with probability one and bounded from below by zero, which implies that this sequence is convergent with probability one to a random variable $\tilde{a}^*$ (see theorem 5.1.1(vii); Rosenlicht, 1968, p. 50).

Assume that $\tilde{a}^*$ is positive and not equal to zero, from equation 5.2,

$$\sum_{t=1}^{\infty} \gamma_t \tilde{a}_t \geq \tilde{a}^* \sum_{t=T}^{\infty} \gamma_t = \infty,$$

which contradicts equation 5.11. Thus, the sequence $\tilde{a}_T^*, \tilde{a}_{T+1}^*, \ldots$ is convergent with probability one to zero. Equivalently a subsequence of $\{|\tilde{\mathbf{g}}_t|^2\}$ is convergent with probability one to zero.

Step 5: Show that the stochastic sequence $\{\tilde{\boldsymbol{\theta}}(t)\}$ *converges to a random variable wp1.* From conclusion (1) of the almost supermartingale lemma, the stochastic sequence of $\ell(\tilde{\boldsymbol{\theta}}(1)), \ell(\tilde{\boldsymbol{\theta}}(2)), \ldots$ converges to some unknown random variable, which will be denoted as $\tilde{\ell}^*$ with probability one. Since ℓ is continuous, this is equivalent to the assertion that $\tilde{\boldsymbol{\theta}}(1), \tilde{\boldsymbol{\theta}}(2), \ldots$ converges with probability one to some unknown random variable, which will be denoted as $\tilde{V}^*$ such that $\ell(\tilde{V}^*) = \tilde{\ell}^*$ with probability one. By the assumption that with probability one, every trajectory $\tilde{\boldsymbol{\theta}}(1), \tilde{\boldsymbol{\theta}}(2), \ldots$ is confined to the closed, bounded, and convex set Θ, it follows that $\tilde{V}^* \in \Theta$ with probability one.

Step 6: Show the stochastic sequence $\{|\tilde{\mathbf{g}}_t|^2\}$ *converges to zero wp1:* Since $\mathbf{g}$ is a continuous function, it follows that $|\mathbf{g}(\tilde{\boldsymbol{\theta}}(1))|^2, |\mathbf{g}(\tilde{\boldsymbol{\theta}}(2))|^2, \ldots$ converges with probability one to $|\mathbf{g}(\tilde{V}^*)|^2$. This is equivalent to the statement that every subsequence of $\{|\mathbf{g}(\tilde{\boldsymbol{\theta}}(t))|^2\}$ converges to $|\mathbf{g}(\tilde{V}^*)|^2$ with probability one. That is, for every possible sequence of positive integers $t_1, t_2, \ldots$ the stochastic subsequence $|\mathbf{g}(\tilde{\boldsymbol{\theta}}(t_1)|^2, |\mathbf{g}(\tilde{\boldsymbol{\theta}}(t_2)|^2, \ldots$ converges with probability one to $|\mathbf{g}(\tilde{V}^*)|^2$.

From step 4, there exists a sequence of positive integers, $k_1, k_2, \ldots$ such that the stochastic subsequence $|\mathbf{g}(\tilde{\boldsymbol{\theta}}(k_1)|^2, |\mathbf{g}(\tilde{\boldsymbol{\theta}}(k_2)|^2, \ldots$ converges with probability one to zero. Thus, to avoid a contradiction, every subsequence of $\{|\mathbf{g}(\tilde{\boldsymbol{\theta}}(t))|^2\}$ converges to a random variable

$|\mathbf{g}(\tilde{V}^*)|^2$ with probability one and additionally with probability one, $|\mathbf{g}(\tilde{V}^*)|^2 = 0$—or equivalently, $\{|\mathbf{g}(\tilde{\boldsymbol{\theta}}(t))|^2\}$ converges to 0 with probability one.

Since $|\mathbf{g}|^2$ is a continuous function and the assumption that $\tilde{V}^* \in \Theta$ with probability one, it follows that $\tilde{\boldsymbol{\theta}}(1), \tilde{\boldsymbol{\theta}}(2), \ldots$ converges with probability one to

$$\{\tilde{V}^* \in \Theta : |\mathbf{g}(\tilde{V}^*)|^2 = 0\}.$$

That is, $\tilde{\boldsymbol{\theta}}(1), \tilde{\boldsymbol{\theta}}(2), \ldots$ converges with probability one to the set of critical points of ℓ in Θ. □

References

Amari, S. (1967). A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, *16*(3), 299–307.

Andrieu, C., Moulines, E., & Priouret, P. (2005). Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization*, *44*, 283–312.

Baird, L., & Moore, A. (1999). Gradient descent for general reinforcement learning. In M. Kearns, S. A. Solla, & D. A. Cohn (Eds.), *Advances in neural information processing systems*, *11*. Cambridge, MA: MIT Press.

Balcan, M. F. F., & Feldman, V. (2013). Statistical active learning algorithms. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Weinberger (Eds.), *Advances in neural information processing systems, 26*. (pp. 1295–1303). Red Hook, NY: Curran. http://papers.nips.cc/paper/5101-statistical-active-learning-algorithms.pdf

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828, http://doi.ieeecomputersociety.org/10.1109/TPAMI.2013.50

Benveniste, A., Metivier, M., & Priouret, P. (1990). *Adaptive algorithms and stochastic approximation*. New York: Springer.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Belmont, MA: Athena Scientific.

Blum, J. R. (1954). Multidimensional stochastic approximation. *Annals of mathematical statistics*, *9*, 737–744.

Borkar, V. S. (2008). *Stochastic approximation: A dynamical systems viewpoint*. New York: Cambridge University Press.

Bottou, L. (1991). Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nîmes 91, EC2*. http://leon.bottou.org/papers/bottou-91c

Bottou, L. (1998). Online algorithms and stochastic approximations. In D. Saad (Ed.), *Online learning and neural networks*. Cambridge: Cambridge University Press. http://leon.bottou.org/papers/bottou-98x.

Bottou, L. (2004). Stochastic learning. In O. Bousquet & U. von Luxburg (Eds.), *Lecture Notes in Artificial Intelligence: Vol. LNAI 3176. Advanced Lectures on Machine Learning* (pp. 146–168). Berlin: Springer-Verlag. http://leon.bottou.org/papers/bottou-mlss-2004

Carbonetto, P., King, M., & Hamze, F. (2009). A stochastic approximation method for inference in probabilistic graphical models. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in neural information processing systems*, *22* (pp. 216–224). Red Hook, NY: Curran. http://papers.nips.cc/paper/3823-a-stochastic-approximation-method-for-inference-in-probabilistic-graphical-models.pdf

Darken, C., & Moody, J. (1992). Towards faster stochastic gradient search. In J. E. Moody, S. J. Hanson, & R. P. Lippman (Eds.), *Advances in neural information processing systems*, *4*. Palo Alto: Morgan Kaufmann.

Delyon, B., M. Lavielle, & E. Moulines (1999). Convergence of a stochastic approximation version of the EM algorithm. *Annals of Statistics*, *27*(1), 94–128. http://dx.doi.org/10.1214/aos/1018031103

Douc, R., Moulines, & E., D. S. Stoffer (2014). *Nonlinear time series: Theory, methods, and applications with R examples*. New York: CRC Press.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In G. J. Gordon, D. B. Dunson, & M. Dudík (Eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (Vol. 15, pp. 315–323). PMLR.

Golden, R. M. (1996). *Mathematical methods for neural network analysis and design*. Cambridge, MA: MIT Press.

Gu, M. G., & Kong, F. H. (1998). A stochastic approximation algorithm with Markov chain Monte-Carlo method for incomplete data estimation problems. *Proceedings of the National Academy of Sciences of the United States of America*, *95*, 7270–7274.

Hazan, E., Levy, K., & Shalev-Shwartz, S. (2015). Beyond convexity: Stochastic quasi-convex optimization. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems*, *28* (pp. 1585–1593). Red Hook, NY: Curran. http://papers.nips.cc/paper/5718-beyond-convexity-stochastic-quasi-convex-optimization.pdf

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*, 1527–1554.

Jaakkola, T., Jordan, M. I., & Singh, S. P. (1994). On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, *6*(6), 1185–1201.

Jani, U., Dowling, E., Golden, R., & Wang, Z. (2000). Multiuser interference suppression using block Shanno constant modulus algorithm. *IEEE Transactions on Signal Processing*, *48*(5), 1503–1506. doi:10.1109/78.840003

Kushner, H. J. (1981). Stochastic approximation with discontinuous dynamic and state dependent noise: w.p. 1 and weak convergence. *Journal of Mathematical Analysis and Applications*, *82*, 527–542.

Kushner, H. (2010). Stochastic approximation: A survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(1), 87–96. http://dx.doi.org/10.1002/wics.57

Kushner, H. J., & Yin, G. G. (1997). *Stochastic approximation algorithms and applications*. New York: Springer-Verlag.

Louis, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society, Series B*, *44*(2), 226–233.

McLachlan, G. J., & Krishnan, T. (1996). *The EM algorithm and its extensions*. New York: Wiley.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. Cambridge, MA: MIT Press.

Ollivier, Y. (2015). Riemannian metrics for neural networks I: Feedforward networks. *Information and Inference*, *4*, 108–153.

Paik, D., Golden, R. M., Torlak, M., & Dowling, E. M. (2006). Blind adaptive CDMA processing for smart antennas using the block Shanno constant modulus algorithm. *IEEE Transactions on Signal Processing*, *54*(5), 1956–1959. doi:10.1109/TSP.2006.870608

Pearlmutter, B. (1992). Gradient descent: Second order momentum and saturating error. In: J. E. Moody, S. J. Hanson, R. P. Lippmann (Eds.), *Advances in neural information processing systems*, *4* (pp. 887–894). San Mateo, CA: Morgan-Kaufmann. http://papers.nips.cc/paper/454-gradient-descent-second-order-momentum-and-saturating-error.pdf

Razaviyayn, M., Hong, M., Luo, Z. Q., & Pang, J. S. (2014). Parallel successive convex approximation for nonsmooth nonconvex optimization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *27* (pp. 1440–1448). Red Hook, NY: Curran. http://papers.nips.cc/paper/5609-parallel-successive-convex-approximation-for-nonsmooth-nonconvex-optimization.pdf

Robbins, H., & Siegmund, D. (1971). A convergence theorem for non negative almost supermartingales and some applications. In J. S. Rustagi (Ed.), *Optimizing methods in statistics* (pp. 233–257). New York: Academic Press.

Rosenlicht, M. (1968). *Introduction to analsis*. New York: Dover.

Roux, N. L., Manzagol, P.-A., & Bengio, Y. (2008). Topmoumoute online natural algorithm. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.), *Advances in neural information processing systems*, *20* (pp. 849–856). Red Hook, NY: Curran.

Roux, N. L., Schmidt, M., Bach, F. R. (2012). A stochastic gradient method with an exponential convergence rate for finite training sets. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *25* (pp. 2663–2671). Red Hook, NY: Curran.

Salakhutdinov, R., & Hinton, G. E. (2012). An efficient learning procedure for deep Boltzmann machines. *Neural Computation*, *24*, 1967–2006.

Schraudolph, N. N., Yu, J., & Günter, S. (2007). A stochastic quasi-Newton method for online convex optimization. In M. Meila, & X. Shen (Eds.), *Proceedings of the 11th Intl. Conf. Artificial Intelligence and Statistics* (Vol. 2, pp. 436–443).

Sugiyama, M. (2015). *Statistical reinforcement learning: Modern machine learning approaches*. London: Chapman & Hall/CRC.

Sunehag, P., Trumpf, J., Vishwanathan, S., & Schraudolph, N. N. (2009). Variable metric stochastic approximation theory. In D. V. Dyk & M. Welling (Eds.), *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*.

Sutskever, I., Marten, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In S. Dasgupta & D. M. Allester (Eds.), *Proceedings of the 30th International Conference on Machine Learning* (pp. 1139–1147).

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Swersky, K., Chen, B., Marlin, B., & de Freitas, N. (2010). A tutorial on stochastic approximation algorithms for training restricted Boltzmann machines and deep belief nets. In *Proceedings of the Information Theory and Applications Workshop* (pp. 1–10). doi:10.1109/ITA.2010.5454138

Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 1064–1071). New York: ACM.

Toulis, P., Rennie, J., & Airoldi, E. (2014). Statistical analysis of stochastic gradient methods for generalized linear models. In *Proceedings of the 31st International Conference on Machine Learning* (pp. 667–675).

Vlassis, N., & Toussaint, M. (2009). Model-free reinforcement learning as mixture learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 1081–1088). New York: ACM. http://doi.acm.org/10.1145/1553374.1553512

White, H. (1989a). Some asymptotic results for learning in single hidden-layer feedforward network models. *Journal of the American Statistical Association*, *84*, 1003–1013. doi:10.1080/01621459.1989.10478865

White, H. (1989b). Learning in artificial neural networks: A statistical perspective. *Neural Computation*, *1*, 425–464.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*, 229–256.

Younes, L. (1999). On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics and Stochastic Reports*, *65*(3), 177–228

Yuille, A. L. (2005). The convergence of contrastive divergences. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems*, *17* (pp. 1593–1600). Cambridge, MA: MIT Press. http://papers.nips.cc/paper/2617-the-convergence-of-contrastive-divergences.pdf

Zhang, S., Choromanska, A. E., & LeCun, Y. (2015). Deep learning with elastic averaging SGD. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems, 28* (pp. 685–693). Red Hook, NY: Curran. http://papers.nips.cc/paper/5761-deep-learning-with-elastic-averaging-sgd.pdf

Zheng, H., Yang, Z., Liu, W., Liang, J., & Li, Y. (2015). Improving deep neural networks using softplus units. In *Proceeding of the 2015 International Joint Conference on Neural Networks* (pp. 1–4). Piscataway, NJ: IEEE. doi:10.1109/IJCNN.2015.7280459

Received August 19, 2017; accepted May 10, 2018.