Sādhanā, Vol. 24, Parts 4 & 5, August & October 1999, pp. 425–452. © Printed in India

Stochastic approximation algorithms: Overview and recent trends

B BHARATH[a] and V S BORKAR[b]*

[a] Department of Electrical Communication Engineering, and
[b] Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India
* Present address: Tata Institute of Fundamental Research, Homi Bhabha Road, Mumbai 400 005, India
e-mail: bharath@protocol.ece.iisc.ernet.in; borkar@tifr.res.in

Abstract. Stochastic approximation is a common paradigm for many stochastic recursions arising both as algorithms and as models of some stochastic dynamic phenomena. This article gives an overview of the known results about their asymptotic behaviour, highlights recent developments such as distributed and multiscale algorithms, and describes existing and potential applications, and other related issues.

Keywords. Stochastic approximation, asymptotic convergence, stochastic optimization, learning algorithms.

1. Introduction

Stochastic approximation is a class of stochastic recursions that goes back to Robbins & Monro (1951). It was pursued with great zeal by statisticians and electrical engineers as a convenient paradigm for recursive algorithms for regression, system identification, adaptive control etc. A comprehensive account of these developments can be found in standard texts such as Benveniste *et al* (1990), Duflo (1997), Kushner & Yin (1997). (Some of the older ones, still of great value, are Wasan 1969, Nevelson & Khasminskii 1976 and Kushner & Clark 1978). The subject has got a fresh lease of life in recent years because of some new emerging application areas. These are broadly covered under the general rubric of 'learning algorithms', encompassing learning algorithms for neural networks, reinforcement learning algorithms arising from artificial intelligence and adaptive control and models of learning by boundedly rational agents in macroeconomics. Consequently, this has thrown up several new issues, both theoretical and practical. The aim of this article is to highlight some of these and to serve as a window to recent developments and as a pointer to the literature for the interested reader.

The archetypical stochastic approximation algorithm is given by the d-dimensional iteration

$$X(n+1) = X(n) + a(n)(h(X(n)) + M(n+1)), \tag{1}$$

where $\{M(n)\}$ is a sequence of random variables interpreted as 'noise' and $\{a(n)\}$ a sequence of positive scalar stepsizes. The two cases of interest will be: $a(n) =$ constant and decreasing $\{a(n)\}$ satisfying

$$\sum_n a(n) = \infty, \quad \sum_n a(n)^2 < \infty. \tag{2}$$

The quantity in parentheses on the right hand side of (1) is interpreted as a noisy measurement of the function $h(\cdot)$ at $X(n)$. The aim is to solve the equation $h(x) = 0$, given a black box (or 'oracle' if you are a computer scientist) that outputs a noise-corrupted measurement of $h(\cdot)$ at the value we input. Ideally, on suitable choice of $\{a(n)\}$, this recursion should converge to a zero of $h(\cdot)$ asymptotically, with probability one.

Here are two motivating examples.

1.1 *Empirical risk minimization*

Suppose we have independent and identically distributed pairs of random variables $(X_i, Y_i), i \geq 1$, and wish to learn a hypothesized relationship: $Y_i \approx f(X_i)$, for f chosen from a family of functions $\{f_\beta\}$, parameterized by a vector β of parameters belonging to a subset A of some Euclidean space. The standard statistical decision theoretic framework for this is to minimize the 'expected risk' $E[l(Y_i, f_\beta(X_i))]$ over $\beta \in A$, where $l(.,.)$ is a prescribed 'risk' function and $E[\cdot]$ stands for the mathematical expectation. For the sake of being specific, consider the special case of 'mean-square error' $E[\|Y_i - f_\beta(X_i)\|^2]$, with $A = R^d$. To minimize this over β, set its gradient w.r.t. β (denoted by $\nabla_\beta(\cdot)$) equal to zero, leading to

$$\begin{aligned} h(\beta) &:= \nabla_\beta E[\|Y_i - f_\beta(X_i)\|^2] \\ &= E[-2\langle \nabla_\beta f(X_i), Y_i - f_\beta(X_i)\rangle] = 0. \end{aligned}$$

(We implicitly assume here that the differentiation and its interchange with the expectation can be justified.) But we cannot evaluate the above expectation if the joint distribution of X_i, Y_i is not known. Taking a cue from (1), we can , however, try the recursion

$$\begin{aligned} \beta_{n+1} &= \beta_n + a(n)\langle \nabla_{\beta_n} f(X_n), Y_n - f_\beta(X_n)\rangle \\ &= \beta_n + a(n)(h(X_n) + M(n+1)), \end{aligned}$$

which specifies the definition of $\{M(n)\}$. This is a special case of (1). The formulation itself, the 'least mean square' or LMS algorithm (see, e.g., Haykin 1994, pp.124–131, also, Gardner 1984) is a special case of the general principle of empirical risk minimization popularised by Vapnik (1995). This has also been among the early motivations for stochastic approximation, particular in the context of adaptive systems (Tsypkin 1971). For some interesting early controversies regarding the validity of this paradigm, see Tsypkin (1966), Stratonovich (1968) and Tsypkin (1968).

1.2 *Nonlinear urn models*

Consider an initially empty urn to which we add red or black balls one at a time, the probability of the ball being red being a function of the current fraction of red balls in the urn. Let $\{y_n\}$, $\{x_n\}$ denote respectively the number and fraction of red balls after n steps and $p : [0, 1] \to [0, 1]$ the function that specifies the probability of the next ball being red

as $p(x_n)$. Then $y_{n+1} = y_n + \xi_n$ where ξ_n is a random variable taking values 0,1. The (conditional) probability of ξ_n being 1 given the present composition of the urn is $p(x_n)$. Dividing through by $(n+1)$, some algebra leads to

$$x_{n+1} = x_n + (1/(n+1))([p(x_n) - x_n] + M(n+1)) \tag{3}$$

with

$$M(n+1) = \xi_n - p(x_n).$$

This is in the form (1) with $h(x) = p(x) - x \; \forall x$.

This dynamics came to the fore in recent years as a model of 'increasing returns' economics (Arthur 1994). Very often one finds that certain new brands or technologies, after some initial randomness, get the dominant share of the market purely through 'positive feedback' mechanisms such as herd instinct and brand loyalties of customers, standardization and so on. Often they seem to get an unreasonable advantage over comparable or even better alternatives (the QWERTY system of typewriters, certain standards in video cassette recorder industry, are among instances cited in this context-see Arthur 1994). This phenomenon can be modelled as a nonlinear urn wherein higher fraction of a colour increases its probability. Other phenomena modelled along these lines are geographical overconcentration of certain industries (the 'silicon valley' syndrome), the evolution of social conventions, and so on.

With these two hopefully appetizing instances, we take up the more technical issues in the following sections. Section 2 discusses issues pertaining to the convergence behaviour of these algorithms. Section 3 describes several variations on the theme, among them distributed implementations and two time scale algorithms. Section 4 surveys several old and new application domains. Section 5 briefly touches upon other related classes of algorithms. Section 6 concludes with some general remarks.

We conclude this section with an interesting insight in the context of (3) above. Note that

$$x_n = \frac{1}{n} \sum_{m=1}^{n} \xi_m. \tag{4}$$

Thus convergence of $\{x_n\}$ would in fact be a law of large numbers for dependent random variables $\{\xi_n\}$. In fact, for any random variables $\{\xi_m\}, \{x_n\}$ defined by (4) can be evaluated recursively as

$$x_{n+1} = x_n + (1/(n+1))(\xi_{n+1} - x_n),$$

which is akin to (1). This link with the (strong) law at large numbers provides some important intuition in the theoretical analysis of (1).

2. Convergence and related issues

2.1 *Convergence analysis*

We first consider the case of $\{a(n)\}$ satisfying (2) above. The two frontrunners among the techniques for convergence analysis of stochastic approximation algorithms are the purely probabilistic techniques, usually based on martingale theory (see, e.g., Ljung *et al* 1992) and the 'o.d.e. approach', originally due to Ljung (1977). A third, purely deterministic

approach based on purely deterministic conditions on the noise sequence has been proposed by Kulkarni & Horn (1996), and Delyon (1996). See also Rachev & Ruschendorf (1995) for an interesting perspective of recursive algorithms as fixed point iterations in the space of probability measures equipped with appropriate metrics.

We shall describe here briefly one variation of the o.d.e (for 'ordinary differential equations') approach, because we shall often invoke the intuition that goes with it in what follows. Extensive accounts of this approach can be found in texts like Benveniste *et al* (1990) and Kushner & Yin (1997). The approach taken here, based on the Hirsch Lemma (Hirsch 1989, pp 339), is a little different.

We shall assume that the noise sequence $\{M(n)\}$ satisfies:

$$E[M(n+1)/M(m), X(m), m \leq n] = 0$$

for all n. Thus,

$$Z(n) = \sum_{i=0}^{n-1} a(i)M(i+1), n \geq 1, \tag{5}$$

satisfies

$$E[Z(n+1)/M(m), X(m), m \leq n] = Z(n), n \geq 1,$$

i.e., $\{Z(n)\}$ is a 'martingale' process. If we assume in addition that $\sup_n E[\|M(n)\|]. < \infty$, the 'martingale convergence theorem' (Borkar 1995, pp.49-50) ensures that $\{Z(n)\}$ converges with probability one. In particular, the 'tail' of the series (5) is asymptotically negligible with probability one, i.e.,

$$\lim_{m,n\to\infty} (Z(m+n) - Z(n)) = 0. \tag{6}$$

Convergence analysis of (1) is often done under the apparently weaker condition (6) or some variant of it, but it is rare to be able to verify (6) without verifying the convergence of $\{Z(n)\}$. Other sufficient conditions have also been considered in the literature. For an overview and comparison, see Wang *et al* (1996).

We shall also assume the stability of the iterates, i.e., that

$$\sup_n \|X(n)\| < \infty, \tag{7}$$

with probability one. This is a nontrival assumption, because this 'stability' is not always easy to verify. We shall comment on this in the next subsection.

The argument we employ hinges on the following mathematical fact: Consider a well-posed ordinary differential equation in R^d:

$$\dot{x}(t) = h(x(t)), \tag{8}$$

assumed to have a globally asymptotically stable attractor J. Let J^ϵ denote ϵ-neighbourhood of J for $\epsilon > 0$. A bounded function $y(\cdot) : [0, \infty] \to R^d$ is said to be a (T, δ)-perturbation of (8) for some $T, \delta > 0$ if one can find $T_n \uparrow \infty$ such that $T_{n+1} - T_n \geq T$ for all n and solutions $x^n(t), t \in I_n := [T_n, T_{n+1}]$ of (8) such that

$$\sup_{t\in I_n} \|x^n(t) - y(t)\| < \delta.$$

Theorem (*Hirsch 1989*). *Given $\epsilon > 0$ and $T > 0$, there exists $\delta_0 > 0$ such that for every $\delta < \delta_0$, a (T, δ)-perturbation $y(\cdot)$ of (8) will converge to J^ϵ.*

The intuition behind this result is as follows: By the converse Liapunov theorem (see, e.g., Wilson 1969), there exists a smooth nonnegative Liapunov function $V(\cdot)$ that strictly decreases along the trajectories of (8) away from J. Thus it will decrease by, say, $\triangle > 0$, along any $X^i(t), t \in I_n$, that does not intersect J^ϵ for a prescribed $n > 0$ sufficiently large. If δ is small, it will decrease by at least $\triangle/2$ along the corresponding patch of $y(\cdot)$, i.e., $y(t), t \in I_n$. This cannot go on forever, so $x^n(\cdot)$ must interest J^ϵ eventually. Some additional work then shows that $x^n(\cdot)$ and $y(\cdot)$ cannot move too much away from J thereafter. See Borkar (1996), and Borkar & Meyn (1997) for a detailed proof.

The way to use this result here is to define an 'interpolated' trajectory of the algorithm as follows: Let $t(0) = 0, t(n) = \sum_{i=1}^{n} a(i), n \geq 1$. Define $\bar{x}(t), t \geq 0$, by $\bar{x}(t(n)) = X(n)$ with linear interpolation on $[T_n, T_{n+1}]$. Then (1) becomes:

$$\bar{x}(t(n+1)) = \bar{x}(t(n)) + h(\bar{x}(t(n)))(t(n+1) - t(n)) + (Z(n+1) - Z(n)). \quad (9)$$

Fix $T > 0$. Let $T_0 = 0$ and $T_{n+1} = min\{t(m) | t(m) \geq T_n + T\}, n \geq 0$. Then $T_n = t(m(n))$ for some increasing sequence $\{m(n)\}$ Let $x^n(\cdot)$ satisfy (8) on $I_n = [T_n, T_{n+1}]$ with $x^n(T_n) = \bar{x}(T_n)$. One may then view (9) as a discretization of the o.d.e. on I_n, with 'noise' $\{Z(i+1) - Z(i), m(n) \leq i < m(n+1)\}$. Standard arguments using the celebrated Gronwall inequality give bounds on

$$\sup_{t \in I_n} \|\bar{x}(t) - x^n(t)\| \quad (10)$$

that depend only on $\max_{m(n) \leq i < m(n+1)} a(i)$ and $\max_{m(n) \leq i < m(n+1)} \|Z(i+1) - Z(m(n))\|$. The former quantity tends to zero with n by our choice of $\{a(n)\}$ and the latter by (6), ensuring that (10) does so as well. Thus $\bar{X}(t + \cdot)$ is a (T, δ)-perturbation of (8) for any $\delta > 0$ if t is chosen large enough (depending, of course, on δ). The 'Hirsch Lemma' above then guarantees that $\bar{x}(t) \to J$, *i.e.*, $X(n) \to J$ with probability one.

Usually, $h(\cdot)$ is chosen so that J is (or at least contains) the desired set of points (e.g., local minima in the case $h = -\nabla f$ described later in this article). Often it is a discrete set of equilibrium points of (8), i.e., $J = \{x | h(x) = 0\}$. In this case, it can be shown that under mild conditions the algorithm avoids unstable equilibria of (8) with probability one (Ljung 1978; Pemantle 1990; Brandiere & Duflo 1996), while it can converge to any of the stable equilibria with positive probability (Arthur *et al* 1983). (Hence the occasional 'undesired equilibria' in nonlinear urn models.) For the general case, not much is known except in special cases. A notable exception to this statement is a recent result of Benaim (1996). Call a point x chain recurrent for (8) if for any $\epsilon > 0$, one can find a chain of points $x_0 = x, x_1, x_2, \ldots, x_n = x$ (say) so that for each i, a trajectory of (8) that starts at x_i comes within ϵ of x_{i+1}. Intuitively these are the points that could be mistaken for periodic points if our measurements were inaccurate. A set is said to be chain recurrent if every point therein is. Benaim shows that under mild conditions, the iterates of (1) will converge to some chain recurrent set with probability one. See Brandiere (1998) and Benaim (1998) for more recent results in this vein. An extensive account of this circle of ideas appears in Benaim (1998).

For the constant stepsize case (i.e., $a(n) = a > 0$ for all n), one cannot hope for convergence with probability one in general. This is because the noise input is 'persistent' as opposed to 'asymptotically negligible' in the decreasing stepsize case (cf. (6) above). In most cases of interest, the theory of Markov processes can be invoked to claim that (1) will

have a limiting stationary distribution. It is then desired that it be concentrated near J. That is, the kind of result one seeks is that for given $\epsilon, \delta > 0$, one can have

$$\limsup_{n\to\infty} P(X(n) \notin J^{\epsilon}) < \delta,$$

for sufficiently small a. This is usually achieved by justifying the interpolated $\bar{x}(\cdot)$ (with $t(n) = na$) as a 'good' approximation to (8). Note that our assumptions on $\{M(n)\}$ imply in particular that they are uncorrelated. Thus if their variance is bounded by $K > 0$ (say), the total noise input to $\bar{x}(\cdot)$ on the interval $[0, T]$, comprising of $\approx T/a$ steps, is of the order of $(T/a) \cdot a^2 \cdot K = KTa$, which goes to zero as $a \to 0$. That is, the smaller the stepsize, the better $\bar{x}(\cdot)$ approximates (8). (Note, however, that the smaller the a, the number of steps to simulate (8) on $[0, T]$ with fixed T grows as T/a. We shall return to this tradeoff later.) See Benveniste *et al* (1987) and Kushner & Yin (1997) for a typical o.d.e. approach to the constant stepsize case.

2.2 *Stability*

Consider the algorithm with $\{a(n)\}$ satisfying (2). As already mentioned, the stability condition (7) does not usually come free. Worse, there is no general purpose methodology for ensuring (7), only a repertoire of special techniques that come with their own limited domains of applicability. The oldest is perhaps the 'stochastic Liapunov function' approach, useful, e.g., for the stochastic gradient schemes described later. We give below an outline of this for a very special and simple case.

Suppose there exists a bounded set $B \subset R^d$ and a nonnegative twice continuously differentiable function $V : R^d \to R$ with bounded second derivatives, satisfying

$$\lim_{\|x\|\to\infty} V(x) = \infty,$$

and

$$\langle \nabla V(x), h(x) \rangle < 0 \text{ for } x \notin B,$$

$V(\cdot)$ is our 'stochastic Liapunov function'. By Taylor's theorem, whenever $X(n) \notin B$, we have

$$\begin{aligned} V(X(n+1)) =& V(X(n)) + a(n)\langle \nabla V(X(n)), h(X(n)) \rangle \\ &+ a(n)\langle \nabla V(X(n)), M(n+1) \rangle + a(n)^2 \xi(n), \end{aligned}$$

for a suitable random variable $\xi(n)$. Thus for $\hat{\xi}(n) = E[\xi(n)/X(m), M(m), m \le n]$,

$$E[V(X(n+1))/X(m), M(m), m \le n] \le V(X(n)) + a(n)^2 \hat{\xi}(n).$$

Under (2) and our assumptions (plus some more) one can claim that $\sum a(n)^2 \hat{\xi}(n)$ is summable. This permits one to use the 'almost supermartingale convergence theorems' (see, e.g., Borkar 1995, pp. 54–55) to claim that $\{X(n)\}$ must hit B once it is outside B. This usually suffices to ensure (7). This is only a crude sketch of a rather sophisticated theory. For more, as well as for variants such as the 'perturbed Liapunov function method', see Kushner (1984).

There are other techniques developed for special classes of algorithms. For example, consider the case $h(x) = f(x) - x$, where $f : R^d \to R^d$ is a contraction w.r.t. a suitable norm

$\|\cdot\|$. That is,

$$\|f(x)-f(y)\| \leq \alpha\|x-y\| \quad \forall x, y,$$

for some $\alpha \in (0,1)$. This situation arises, e.g., in some reinforcement learning algorithms for dynamic programming which we discuss later in this article. Under this condition and other mild conditions (e.g., on noise), Tsitsiklis (1994) proved (7).

Another approach for a slightly larger class of problems arising out of similar applications looks at iterates of the type

$$X(n+1) = G_n(X(n), M(n+1)),$$

where $G_n(\cdot,\cdot)$ satisfy: $\forall x, y, z$,

$$\|G_n(x,y) - G_n(z,y)\| \leq \|x-z\|,$$

and

$$G_n(\alpha x, y) = \alpha G_n(x,y) + (1-\alpha)C(y), \alpha \in (0,1),$$

with $|C(y)|$ uniformly bounded in y. In addition, suppose that the above iteration, modified so that it is reset to a fixed value in a large ball centred at the origin each time it exists from it, converges to some point therein with probability one. Then one can show that the original iterates satisfy (7). The proof is based on first proving that if the iterates satisfy (7) for one initial condition they do so for any initial condition. Since the iterates with resets converge as specified, there are at most finitely many resets, implying (7) for some initial condition, therefore for all initial conditions. This idea, which covers certain learning algorithms for dynamic programming, is from Jaakola *et al* (1994) and was further developed in Abounady *et al* (1996a), which, in fact, does away with the second condition on G_n above.

A more recent approach for the same class of algorithms uses the following property for the function h arising therein: $\bar{h}(x) = \lim_{a\to\infty} h(ax)/a$ exists and the o.d.e.

$$\dot{x}(t) = \bar{h}(x(t)), \tag{11}$$

has a globally asymptotically stable equilibrium x^*. (It is not hard to see that this will have to be the origin.) One mimics the argument used for convergence analysis in the preceding subsection with some crucial differences. First, the interpolated trajectory $\bar{x}(\cdot)$ is rescaled on each interval I_n so as to have a uniformly bounded value at the beginning of the interval. Second, one works with the Liapunov function $V(\cdot)$ associated with (11) rather than (8), and rather than show $V(\bar{x}(T(n+1))) - V(\bar{x}(T(n)))$ is negative for $\bar{x}(T(n))$ outside some set, one shows that the ratio $V(\bar{x}(T(n+1)))/V(\bar{x}(T(n))) < 1/2$ when $\bar{x}(T(n))$ is outside some set. This suffices to ensure (7). This proof, from Borkar & Meyn (1997), is perhaps the first instance of the o.d.e. method being used to prove both stability and convergence, rather than only the latter.

Finally, one can simply cheat out of the stability issue by projecting the iterates back onto a prescribed large bounded set B presumed to contain J whenever they exit from the same. The o.d.e. tracked by the projected algorithm is, however, no longer (8) but a modification thereof that modifies h on the boundary of B so as to keep trajectories originating in B inside B (Kushner & Clark 1978). Though this avoids the stability issue, it can lead to spurious, undesired attractors on the boundary of B for the o.d.e. and hence for the algorithm.

A novel scheme of Chen (1994) truncates the iterates and gradually increases the truncation levels to ensure stability. See also Chen (1998b) and Tadic (1998). For the constant stepsize case, (7) cannot be expected to hold in general unless the iterates are being projected back to a bounded set. The correct stability concept is the requirement that the laws of $X(n), n \geq 0$, remain 'tight', i.e. for any $\epsilon > 0$, one can find an $N_\epsilon \geq 1$ such that

$$P(\|X(n)\| \geq N_\epsilon) < \epsilon \ \forall n.$$

If $\{X(n)\}$ is Markov or function of a Markov process, this amounts to looking for asymptotic stationarity. Stability theory in this sense has been extensively studied and an indepth account can be found in Meyn & Tweedie (1994).

2.3 *Convergence rates*

Recall the connection between the decreasing stepsize algorithm and the strong law of large numbers. For independent and identically distributed random variables $\{X_i\}$ with finite variance σ^2, the strong law says that the arithmetic mean $S_n := \frac{1}{n}\sum_{i=1}^{n} X_i$ approaches the expectation $E[X_1]$ in the limit with probability one. The fluctuations around this 'typical' behaviour, given by $S_n - E[X_1]$, is quantified in an asymptotic sense by the central limit theorem which states that $\sqrt{n}(S_n - E[X_1])$ converges in distribution to a Gaussian distribution with mean zero and variance σ^2. In this sense, one may say that the convergence rate for the strong law of large number is $O(n^{-1/2})$.

This philosophy can be extended to the stochastic approximation algorithm. Consider the decreasing stepsize as in (2) to start with. Let $\tilde{x}^n(t), t \geq t(n)$, be the solution to (8) with $\tilde{x}(t(n)) = X(n)$. As before one can show that for any $T > 0$,

$$\sup_{t \in [0,T]} \|\bar{x}(t(n) + t) - \tilde{x}^n(t(n) + t)\| \to 0,$$

as $n \to \infty$. Thus (8) captures the 'typical' behaviour of (1). The 'fluctuation' part is $\bar{x}(t(n) + \cdot) - \tilde{x}^n(t(n) + \cdot)$, $n \geq 0$. One can show a 'functional central limit theorem' (also known as an 'invariance principle') to the effect that $(\bar{x}(t(n) + \cdot) - \tilde{x}^n(t(n) + \cdot))/(a(n))^{1/2}$ converges in distribution to a Gauss–Markov process, i.e., a linear system driven by a Gaussian noise. In turn, in the convergent case (i.e., $X(n) \to x$ where x is the unique asymptotically stable equilibrium for (8)), $(X(n) - x)/(a(n))^{1/2}$ converges in distribution to a zero mean Gaussian, corresponding to the stationary distribution of the Gauss–Markov process. This can be viewed as a 'rate of convergence' result for (1).

In the constant stepsize case, analogous results are available in the limit as the constant stepsize 'a' tends to zero. Both these cases are extensively dealt with in Kushner & Yin (1997).

These results, however, only capture the convergence in distribution and are not sample path-wise. Some intuition about the sample path behaviour can be gleaned from our convergence proof in § 2.1. The o.d.e. (8) will have a certain convergence rate (say, exponential) to $J^\epsilon, \epsilon > 0$. One expects the interpolated trajectory $\bar{x}(\cdot)$ to mimic it eventually. There is, however, a time scaling $n \to t(n)$ involved (which is, e.g., logarithmic for $a(n) = 1/n$), which needs to be inverted to infer the (slower) convergence rate of the original algorithm. This intuition, however, is of limited applicability because one does not know the random time after which the algorithm can be deemed to have been locked into step with (8). For exact asymptotic pathwise rates for a limited class of algorithms, see Chen (1998), Ljung

et al (1992). An exciting recent development is a 'law of iterated logarithms' for stochastic approximation algorithms which captures its almost sure behaviour in a precise way-see Pelletier (1998).

Returning to the connection with the law of large numbers, there is yet another limit theorem associated with the quantity S_n. This is the 'large deviations' result of Cramer which captures the rate of exponential decay of the probability $P(S_n \in A)$ for A bounded away from $E[X_1]$. 'Functional' forms of this result are also available, the most relevant for our purposes being the Freidlin–Wentzell theory of large deviations associated with the 'small noise limit' (Freidlin & Wentzell 1984). A stochastic process described as a noise-driven dynamical system converges in an appropriate sense to the deterministic trajectory of the corresponding noise free dynamics as the noise level is decreased to zero. This is the 'typical' behaviour. Freidlin–Wentzell theory quantifies the relative probabilities of 'untypical' events in the small noise limit. This has been used effectively by Dupuis & Kushner (1985,1989) and Dupuis (1988) to study the 'fine' behaviour of stochastic algorithms.

2.4 *Variance reduction*

Stochastic approximation algorithms are notorious for high variance, reflected in highly oscillatory trajectories, particularly in the initial stages. Much work has gone into variance reduction techniques that use special features of the problem at hand. Traditional techniques for simulation-based algorithms, such as use of common randomness, control variates, importance sampling etc. have been extensively dealt with in standard textbook treatments such as Ripley (1987) and Rubinstein (1981). There are also works on optimal stepsize selection (Ruppert 1988), transformation (preprocessing) of data (Anbar 1973; Abdelhamid 1973) etc. A technique which has attracted much attention recently is that of averaging due to Polyak (1991); Polyak & Juditsky (1992). (See also Yin 1981, Yin & Yin 1994.) The idea is to augment (1) by

$$Y(n) = \frac{1}{n}\sum_{m=1}^{n} X(m),$$

recursively computable by

$$Y(n+1) = Y(n) + \frac{1}{n+1}(X(n+1) - Y(n)).$$

Polyak shows that this is asymptotically optimal in the sense that the variance of the limiting Gaussian distribution for scaled fluctuations (cf. § 2.3 above) attains the theoretical minimum. Kushner & Yang (1995) study an 'on line' version of this scheme which replaces (1) by

$$X(n+1) = X(n) + a(n)(h(Y(n)) + M(n+1)).$$

Weighted averaging has also been studied (Dippon & Renz 1997).

The folk wisdom of this subject speaks of a 'bias-variance dilemma', which makes sense in the finite time horizon situation. Consider an instance of (1) that asymptotically converges to a desired x^* with probability one. In reality, we shall have a finite run of, say, N iterations. This will have two kinds of errors: the average behaviour (over several such runs) will be away from x^* introducing 'bias'. There will be an additional error around this

bias due to data-dependent fluctuations – the 'variance'. The 'dilemma' says that beyond a point one cannot reduce one of these without increasing the other. Thus the variance reduction techniques described above will generally increase the bias. The dynamical picture gives a clue as to how the averaging techniques effectively introduce 'inertia' or 'friction' which, while reducing fluctuations, slows down the net movement towards the desired goal. This intuition also applies to the stepsize selection. Large stepsizes lead to faster movement. For example, in the constant stepsize case, the algorithm simulates (8) on $[0, T]$ by using $\approx T/a$ steps, a being the stepsize. Thus smaller a means more steps to go as far. But the flip side of this is that the total noise variance (assuming constant variance of $M(n), n \geq 1$, say, σ^2) in this interval is $\approx (T/a) \cdot a^2\sigma^2 = aT\sigma^2$, which grows with a. This trade-off is captured by a bound for asymptotic error variance in Borkar & Meyn (1997) which decreases exponentially to a constant with increasing number of steps, where this constant decreases to zero as the stepsize is reduced to zero. But the exponent for this exponential decay deteriorates with decreasing stepsize, vanishing in the limit as the stepsize approaches zero.

3. Variations

3.1 *Distributed stochastic approximation*

Many applications require that the algorithm (1) be implemented in a distributed fashion. That is, each component is updated by one dedicated processor but two distinct components need not be updated by the same processor. Also, the updating by different processors need not be synchronized.

The earliest analysis of such an algorithm is perhaps the work of Tsitsiklis *et al* (1986) for the special case of the stochastic gradient algorithm described in the next section. (See also Bertsekas & Tsitsiklis 1989.) A somewhat more general situation was analysed in Kushner & Yin (1987). Our account is based on Borkar (1996, 1998) which gives a very comprehensive formulation of the problem that clearly underscores the role of asynchronism, communication delays etc.

It is convenient to consider (1) in the form (though more general situations can be handled – see e.g., Konda & Borkar 1999)

$$X(n+1) = X(n) + a(n)f(X_1(n), \ldots, X_d(n), \xi(n)), \tag{12}$$

where $X_i(n), 1 \leq i \leq n$, are components of $X(n)$ and $\{\xi(n)\}$ are i.i.d. Thus $h(\cdot)$ is given by

$$h(x) = E[f(X(n), \xi(n))/X(n) = x],$$

and $\{M(n)\}$ are defined correspondingly. We call (12) the 'centralized' algorithm. There are two formulations of the decentralized or 'distributed' version. The first is the synchronous case in which the ith component is updated as per

$$X_i(n+1) = X_i(n) + a(n)f(X_1(n - \tau_{i1}(n), \ldots, X_d(n - \tau_{id}), \xi(n))I\{i \in \overline{Y}(n)\}, \tag{13}$$

where $\{\tau_{ij}(n)\}$ are random delays encountered in receiving the output of jth processor by the ith processor, $\{\overline{Y}(n)\}$ is a set-valued process taking values in the subsets of $\{1, 2, \ldots, d\}$ that identifies the components that will be updated at time n, and $I\{i \in \overline{Y}(n)\} = 1$, if $i \in Y(n)$ (i.e., ith component is updated at time n) and 0 otherwise.

This is synchronous because it presupposes a 'global clock' identified with the discrete time index $\{n\}$, that is known to all processors. In the asynchronous version, we do away with this requirement. We also allow the stepsize to depend on the component. Thus ith component uses the stepsize schedule $\{a(n,i), n \geq 0\}$, which we assume satisfies (2). Let

$$\nu(i,n) = \sum_{m=0}^{n} I\{i \in \overline{Y}_m\}, \quad 1 \leq i \leq d, n \geq 0.$$

This is the number of times component i was updated till time n and is thus known to the ith processor. The algorithm then is: for $1 \leq i \leq d$,

$$X_i(n+1) = X_i(n) + a(\nu(i,n),i) f(X_1(n - \tau_{i1}(n)), \ldots, X_d(n - \tau_{id}(n)), \xi(n)) I\{i \in \overline{Y}(n)\}. \tag{14}$$

As for delays, we assume that $\tau_{ii}(n) = 0$ for all i and $\{\tau_{ij}(n)\}$ are bounded by a common $\bar{\tau} > 0$. (This can be relaxed.) The process $\{\bar{Y}(n)\}$ is assumed to be such that

$$\liminf_{n \to \infty} \frac{\nu(i,n)}{n} \geq a > 0 \ \forall i, \tag{15}$$

for some deterministic constant $a > 0$. That is, all components are updated comparably often.

Algorithm (13) and (14) can be analysed under these and a few more technical conditions. In particular, (14) needs extra restrictions on the stepsize $\{a(n,i)\}$, the most important of which is

$$\lim_{n \to \infty} \left\{ \sum_{m=1}^{[\alpha n]} a(m,i) \right\} \Big/ \left\{ \sum_{m=1}^{n} a(m,i) \right\} = 1,$$

where $\alpha \in (0,1)$ and $[\alpha n]$ = the largest integer not exceeding αn. The upshot of the analysis, which closely mimics that of § 2.1 , is that if (7) holds with probability 1, then an appropriately interpolated trajectory of the algorithm tracks the nonautonomous o.d.e.

$$\dot{x}(t) = M(t) h(x(t)), \tag{16}$$

where for each $t, M(t)$ is a diagonal matrix with nonnegative diagonal elements that add to one. (This result does not really need (15), but (15) is needed for any subsequent analysis that tries to show that, possibly under further conditions, (16) has qualitative behaviour similar to that of (8).)

One of the two 'complications' we introduced, viz., the communication delay, plays no role here. The reason becomes apparent on a closer scrutiny of the arguments of § 2.1. Note that the time scaling $n \to t(n)$ (logarithmic in case of $a(n) \sim 1/n$) has a decreasing slope and thus in the passage from the original clock $\{n\}$ to the distorted clock $\{t(n)\}$, time intervals $[t, t+T]$ for a fixed $T > 0$ get mapped into smaller and smaller time intervals as t increases, getting completely 'squeezed out' as $t \to \infty$. Thus bounded delays contribute nothing to the asymptotic behaviour. The second complication is that $Y(n)$ need not be all of $\{1, 2, \ldots, d\}$. Writing $\mu(t) = \text{diag}(\mu_1(t), \ldots, \mu_d(t))$, $\mu_i(t)$ can then be interpreted as the instantaneous relative frequency with which i is being updated, after the time scaling. Condition (15) then ensures that $\mu_i(t), 1 \leq i \leq d, t \geq 0$, remain bounded away from zero

from below. In many cases (e.g., when $h(x) = f(x) - x$ where $f(\cdot)$ satisfies

$$\|f(x) - f(y)\| \leq \|x - y\|$$

for a suitable norm) this suffices to ensure desired asymptotic behaviour (Borkar 1996b).

For constant stepsize algorithms, similar analysis is possible in the $a \to 0$ limit. For fixed $a > 0$, however, very complex behaviour is possible even in linear iterations with stationary delays, as observed by Chazan & Miranker (1969) and analysed in great detail by Gharavi & Ananthram (1999).

3.2 *Two-timescale algorithms*

Consider the coupled iterations

$$X(n+1) = X(n) + a(n)(f(X(n), Y(n)) + M(n+1)), \tag{1}$$

$$Y(n+1) = Y(n) + b(n)(g(X(n), Y(n)) + M'(n+1)), \tag{2}$$

where both $\{a(n)\}, \{b(n)\}$ satisfy (2) with $a(n)/b(n) \to 0$ and $\{M(n)\}, \{M'(n)\}$ are noise sequences as before. The intuition here is that the two components of the iterations $X(\cdot)$ and $Y(\cdot)$ use different stepsizes so that one (the $Y(n)'s$) moves faster than the other. The fast component $\{Y(n)\}$ then sees the slow component $\{X(n)\}$ as almost static, while the latter sees the former as 'almost equilibrated'. This intuition can be made precise by analysis similar to that of § 2.1. Suppose that for fixed x, the o.d.e.

$$\dot{y}(t) = g(x, y(t)),$$

has an asymptotically stable equilibrium $\lambda(x)$, where $\lambda(\cdot)$ is a nice function, and the o.d.e.

$$\dot{x}(t) = f(x(t), \lambda(x(t))),$$

has an asymptotically stable equilibrium x^*, then a suitably interpolated version of $Y(\cdot)$ 'tracks' $\{\lambda(X(n))\}$ while $X(n)$ converges to x^* with probability one. Thus, the pair $(X(n), \lambda(X(n)))$ converges to $(x^*, \lambda(x^*))$ with probability one (under suitable stability and regularity conditions). The situation is analogous to the 'singular differential equation'

$$\dot{x}(t) = f(x(t), y(t)),$$
$$\epsilon \dot{y}(t) = g(x(t), y(t)),$$

in the $\epsilon \downarrow 0$ limit.

The way to use this is as follows: Many algorithms have two (or more) nested loops, where the inner loop is itself another full-fledged algorithm (a 'subroutine') whose near-convergence has to be awaited between every two consecutive iterates of the outer loop. The foregoing allows us to achieve the same effect by running both loops concurrently, but with different time scales as reflected in different choices for the stepsize schedule. The inner loop must move faster than the outer one.

The two-timescales idea is not new, but was used originally purely as a device for damping oscillation in the spirit of 'momentum' methods of optimization (Fogel 1981; Ruszczynski & Syski 1983). A more complete treatment was given in Borkar (1996c) and novel applications to simulation-based optimization were proposed and carried out later in Bhatnagar & Borkar (1998) and Konda & Borkar (1999).

One should keep in mind, however, that though $a(n)/b(n) \to 0$ suffices for the above analysis, in practice the two time scales should be sufficiently separated so that fluctuations at one scale do not completely dominate the other.

Analogous analysis is possible for constant stepsize algorithms with $a(n) = a, b(n) = b$ for all n, in the limit as $a, b \to 0$ with $a/b \to 0$.

A different use of two time scales appears in the work of Gelfand & Mitter (1991) which is the discrete time, continuous state space analog of the simulated annealing algorithm (see e.g., Desai 1999). There, $h(\cdot) = -\nabla f(\cdot)$ where $f(\cdot)$ is the function to be minimized, and the algorithm is

$$X(n+1) = X(n) - a(n)(h(X(n)) + M(n+1)) + (a(n))^{1/2} b(n) W(n),$$

where $\{W(n)\}$ is i.i.d. noise deliberately added to (1) to prevent the algorithm from getting stuck in local minima.

The idea then is to let $b(n)$ tend to zero at a much slower rate than $a(n)$, so that a suitably interpolated version of the iterates asymptotically tracks (in the sense of distribution) the stochastic differential equation,

$$\mathrm{d}X(t) = -\nabla f(X(t))\mathrm{d}t + \epsilon^2(t) dB(t),$$

where $B(\cdot)$ is a standard Brownian motion in R^d and $\epsilon^2(t) \to 0$ as $t \to \infty$. This will be recognised as the Langevin algorithm (Gidas 1985) which is the continuous time-continuous space analog of simulated annealing. This is known to converge in probability to the set of global minima. Exactly the same behaviour can then be inferred for the Gelfand–Mitter scheme. For analysis of such iterations where $h(\cdot)$ is not necessarily a gradient, see Fang *et al* (1997).

Constant stepsize analogs of this $(a(n) = a, b(n) = b \ll a)$ have been considered in Pflug (1986) and Borkar & Mitter (1997) where the latter allows for more general $h(\cdot)$ and gets an approximation result in the 'strong' sense of Kullback–Leibler divergence. It is, however, of greater interest to analyse the situation when $a \to 0, b = \delta(a) \to 0$ and find the correct choice of $\delta(\cdot)$ to achieve the desired asymptotic behaviour. A limited success in this problem has been reported (Pflug 1996; S Gelfand 1996, personal communication).

In this context, see also the interesting recent work of Basak *et al* (1997).

3.3 *Passive stochastic approximation*

The model underlying (1) presupposes that we have at our disposal a black box to which we choose to input $X(n)$ at time n and get as output $(h(X(n)) + M(n+1))$ as a noisy measurement of $h(X(n))$. There may, however, be situations where we do not choose the inputs $\{X(n)\}$, but they arise out of some random device on which we have no control. This situation was considered by Hardle & Nixdorf (1987), who proposed the following scheme, called 'passive stochastic approximation'.

$$Z(n+1) = Z(n) + \frac{a(n)}{b(n)} K\left(\frac{X(n) - Z(n)}{b(n)}\right)(h(X(n)) + M(n+1))$$

where $\{a(n)\}, \{b(n)\}$ are deterministic sequences chosen by the algorithm designer, as is the 'kernel' $K(\cdot)$ which is often a 'bell-shaped curve'. This algorithm thus combines features of the 'kernel estimation' methods of nonparametric statistics and stochastic

approximation. See Nazin *et al* (1990), Yin & Yin (1996) and the references therein for further work in this direction.

4. Applications

4.1 *Optimization*

The simplest stochastic optimization algorithm of stochastic approximation type is the stochastic gradient scheme wherein $h(\cdot) = -\nabla f(\cdot), f(\cdot)$ being the function to be minimised. Usually $f(\cdot)$ itself serves as a Liapunov function for the corresponding o.d.e.

$$\dot{x}(t) = -\nabla f(x(t)),$$

ensuring convergence to the set of local minima with probability one. (See Bertsekas & Tsitsiklis 1999, for a detailed convergence analysis under very general conditions.)

This, however, presupposes that a noisy measurement of the gradient is available. This is often not the case. Then, one is obliged to estimate the gradient based on, say, noisy function evaluations. The simplest such scheme is the Kiefer–Wolfowitz scheme (Kiefer & Wolfowitz 1952), which uses a finite difference approximation for each component of the gradient. Thus one needs two function evaluations: one at the nominal value of the parameter and another at a suitable perturbation of it, per component of the gradient. For large dimensional problems, computational overheads of these evaluations can be high. This has led to alternative schemes based on finite difference approximation in randomized directions (Spall 1992).

More generally, one is content to have the net movement of the algorithm not strictly along the negative gradient, but along a 'descent' direction, i.e., a direction that makes an acute angle with the negative gradient. (Note that if $h(\cdot)$ is a descent direction in (8), $f(\cdot)$ can still be a Liapunov function.) This has prompted several 'pseudogradient' methods, modelled after their deterministic counterparts in optimization theory. (See Polyak & Tsypkin 1973, for an extensive account and further references.) In particular, classical techniques from deterministic optimization for improving performance of gradient schemes find their counterparts in stochastic optimization. By way of examples, see Ruszczynski & Syski (1983) for stochastic conjugate gradient-like schemes and Ruppert (1985) for a stochastic Newton–Raphson scheme.

Another pseudo-gradient scheme worth noting is that due to Fabian (1960) where each component of $h(\cdot) = -\nabla f(\cdot)$ in (1) is replaced by the sign thereof. This is particularly appealing for distributed implementation, because one needs only one bit of information per component. Also, replacing a pseudogradient by its componentwise sign leads to a more graceful (i.e., low oscillations) behaviour in the early stages at the expense of slower speed in the long run, as was observed by Bharath & Borkar (1998). This is, of course, fully in accordance with the bias-variance dilemma. These observations suggest the following 'graded quantization' scheme for managing the bias-variance trade-off. Consider several possible quantizations of the components of the pseudogradient, ranked by the number of quantization levels. At one extreme is the Fabian scheme with just two levels, at the other is the full pseudogradient, corresponding to no quantization at all. The idea is to start with the Fabian scheme to suppress the highly oscillatory behaviour in the initial stages and then gradually increase the number of quantization levels to find a good balance between variance suppression and speed of the algorithm.

One scheme that avoids any kind of direct gradient estimation altogether is that due to Katkovnik & Kulchitsky (1972). The idea is as follows. Suppose we replace ∇f by its

approximation

$$Df_\sigma(x) = \int G_\sigma(x-y) \bigtriangledown f(y)\mathrm{d}y \text{ componentwise}$$

where G_σ is a Gaussian with zero mean and variance σ^2. For small σ^2, this is a good approximation to $\bigtriangledown f(x)$. Integrating by parts, we have

$$\begin{aligned} Df_\sigma(x) &= \int \bigtriangledown G_\sigma(x-y) f(y)\mathrm{d}y \\ &= \int G_\sigma(x-y)\tilde{f}_\sigma(\cdot), \end{aligned}$$

for a suitably defined $\tilde{f}_\sigma(\cdot)$, after rearranging terms. But the last expression is simply the expectation of $\tilde{f}_\sigma(x+\xi)$ for $\xi \sim N(0,\sigma^2)$. This leads to an estimate of Df_σ via a Monte Carlo simulation given by

$$\frac{1}{\sigma}\frac{1}{N}\sum_{i=1}^{N} \eta_i \tilde{f}_\sigma(x+\eta_i\sigma),$$

where $\eta_i's$ are i.i.d., $N(0,1)$. Bharath & Borkar (1998) consider a two-timescale scheme where this averaging is done concurrently on a fast time scale and interface it with a Fabian-like two level quantization to suppress the numerical instabilities caused by the small σ on the denominator above. One advantage of these schemes when used for simulation-based optimization is that they need a single simulation.

The number of simulations required is of major concern in simulation-based parametric optimization of large systems. For the special case of 'discrete event dynamical systems', this has led to estimation of gradient based on a single simulation trajectory and its perturbations (Ho & Cao 1991). This is the so-called 'perturbation analysis' (PA) which has led to many other variants such as 'infinitesimal perturbation analysis' (IPA) (Ho *et al* 1983), 'finite perturbation analysis' (FPA) (Ho *et al* 1983), 'smooth perturbation analysis' (SPA) (Gong & Ho 1987), extended perturbation analysis (EPA) (Ho & Li 1988) and so on. These schemes often have a stochastic approximation-like component, e.g., the scheme of Chong & Ramadge (1994) which estimates the gradient based on an explicit stochastic representation of the dependence of observations on the parameter and obtained by aggregating data between regeneration epochs.

In many situations, including some of the above, one seeks the gradient of an averaged 'performance measure' and its interchange with the average of the gradient is either difficult to implement or not justified. Examples arise in certain parametric optimization problems for hidden Markov models. In such cases, the two-timescale paradigm can be used by performing the averaging on a faster time scale (Bhatnagar & Borkar 1998; Bharath & Borkar 1998).

4.2 *Economics and game theory*

In introduction, we already mentioned the nonlinear urn model for 'increasing returns' phenomenon in macroeconomics. There are other models of learning by individual macroeconomic agents that are stochastic approximation-type recursions: See, e.g., Sargent (1993). The appealing features of these dynamics from the economists' point of view are:

(i) *Bounded rationality* – The dynamics uses simple, local adjustments rather than exact optimization. The latter would presuppose, as was done in neoclassical economics, a perfectly rational agent with perfect information and infinite computational capacity and memory. The stochastic approximation-based learning models assume 'bounded rationality': The agents make only simple, intuitive decisions that require small amounts of computation or memory.

(ii) *Incrementality* – The agents in these learning models adapt in small, incremental steps due to inertia, risk aversion etc., which again conforms to one's intuition about real economic agents.

(iii) *Noisy evolution* – These dynamics are also noisy, capturing the effects of noisy information or mistakes and/or deliberate noisy probing by the agents.

It is worth noting that by their inherent nature, these dynamics model 'distributed learning' with possibly many time scales and therefore the theories of § 3.1 and 3.2 should be of much relevance here.

In microeconomics, game theory provides much of the foundation. In non-cooperative games, the natural and accepted notion of equilibrium is that of Nash equilibrium, wherein no agent can unilaterally make a move to improve his lot, all else being kept constant. The problem usually is that there are too many candidate Nash equilibria and one wishes to identify one (or a few) as being more 'natural' than others. These considerations led to years of work on refinement of the Nash equilibrium concept, culminating in the monumental work of Harsanyi & Selten (1988). More recently, however, there has been a shift towards dynamic models of individual learning, wherein individual agents adapt their strategies based on experience. The idea is then to look for asymptotic equilibria for these dynamics that are stable under a variety of perturbations (including stochastic) and hope that these lead to a unique choice of the Nash equilibrium among many.

The oldest such model is that of a 'fictitious play' (Brown 1951) wherein each agent responds to the empirically observed behaviour of the rest. Consider a two-person game with strategy set $\{1,2,\ldots,d\}$ and payoff matrix $A=[[a_{ij}]]$. Suppose that 1 and 2 play mixed strategies $p(t)=[p_1(t),\ldots,p_d(t)]$, $q(t)=[q_1(t),\ldots,q_d(t)]$ respectively at time t. That is, 1 (resp. 2) plays strategy i with probability $p_i(t)$ (resp, $q_i(t)$). Suppose that when 1 (resp. 2) plays the mixed strategy p, the unique best response of 2 (resp. 1) is $f_1(p)$ (resp. $f_2(p)$) for a 'nice' function $f_1(\cdot)$ (resp, $f_2(\cdot)$). Let $\xi_1(t)$ denote the actual strategy played by agent $i, i=1,2$, at time t and define empirical probabilities for agent i by: For $t\geq 0$, $\nu^i(t)=[\nu_1^i(t),\ldots,\nu_d^i(t)], i=1,2$, where $\nu_k^i(t)=(1/(t+1))$ (no. of times agent i played strategy k up to time t). Then,

$$\nu_k^i(t+1)=\nu_k^i(t)+(1/(t+1))(I\{\xi_i(t)=k\}-\nu_k^i(t)),$$

for $1\leq k\leq d$, $i=1,2$, where $I\{\xi_i(t)=k\}=1$ if $\xi_i(t)=k, 0$ otherwise. Furthermore,

$$\begin{aligned}E[I\{\xi_i(t)=k\}/\nu^1(s),\nu^2(s),s\leq t] &= P[\xi_i(t)=k/\nu^1(s),\nu^2(s),s\leq t]\\ &= [f_i(\nu_j(t))]_k,\ j\neq i.\end{aligned}$$

The above iteration is of the form (1) and thus can be expected to track asymptotically the o.d.e.

$$\dot{\nu}^1(t) = f_1(\nu_2(t)) - \nu_1(t) := g_1(\nu^1(t), \nu^2(t)),$$
$$\dot{\nu}^2(t) = f_2(\nu_1(t)) - \nu_2(t) := g_2(\nu^1(t), \nu^2(t)).$$

(More generally, if the 'best response' is not unique, one is led to a differential inclusion.) Note that $\partial g_i(x_1, x_2)/\partial x_j = -1 < 0$ for $i \neq j$, characterizing this o.d.e. as 'competitive'. This fact has been used to advantage by Benaim & Hirsch (1997) to give a complete analysis of this scheme in a very special simple case.

In general, however, this apparently simple dynamics can lead to many complicated situations. For example, the two agents may play the right strategy on the average, but may be 'out of step' with each other so that the actual payoffs they receive are very low. For this and other subtleties, see Fudenberg & Kreps (1993). For an important recent contribution, see Kaniovski & Young (1995). Young (1998) gives an excellent overview of the general field.

A multi-agent variant of this model has been considered in Thathachar & Sastry (1987) in the context of a specific application domain.

There are several other models of learning in games, not necessarily of stochastic approximation variety, that go under the general rubric of 'evolutionary games'. See the recent texts by Vega-Redondo (1996) and Fudenberg & Levine (1998) for a comprehensive treatment. We conclude this subsection with the remark that even 'replicator dynamics', the differential equation model from evolutionary biology that models evolution of species under selection pressure (see, e.g., Hofbauer & Sigmund 1988), has been 'recovered' as a limiting case of a model of individual learning that employ stochastic approximation-like dynamics (Borgers & Sarin 1997).

4.3 *Adaptive systems*

Many problems in control and communication engineering call for tracking a trajectory with unknown statistics or learning system parameters 'on line'. Many algorithms developed for such purposes, which constitute the field of 'adaptive filtering', are stochastic approximations. Extensive textbook accounts of these are now available (Widrow & Stearns 1985; Haykin 1991; Solo & Kong 1995), to which we point for details. We shall confine ourselves to some comments of practical relevance. In applications involving tracking of slowly varying parameters, decreasing stepsizes as in (2) are no use because eventually the algorithm becomes slower than the slowly varying parameter it is supposed to track (unless some 'reset' mechanism is used, complicating the algorithm). For this reason, constant stepsize is preferred. Sophisticated algorithms may use adaptive stepsizes. When the algorithm is implemented in hardware, constant stepsize is usually preferred because of its simplicity.

In the related area of system identification, one tries to learn the system parameters based on observed data. Many algorithms for this (prediction error, recursive least squares, maximum likelihood, ...) tend to be akin to (1), except that the $a(n)$'s may now be random. Excellent treatments of these appear in Ljung & Soderstrom (1983) and Ljung (1986).

In adaptive control, one is trying to control a dynamical system whose dynamical law is unknown. Many schemes for adaptive control hypothesize a parametric model and tune the parameters and control simultaneously 'on line'. (Examples are self-tuning control, model reference adaptive control etc.) For a recent reference that deals with linear stochastic

systems, see Chen & Guo (1991). Algorithms in similar vein for control of finite Markov chains can be found in El Fattah (1981) and Borkar (1996a).

4.4 *Neural networks*

Many, if not most, algorithms for training neural networks are special instances of stochastic approximation, most of them variants of our example 2 from § 1. Since the mainstream algorithms are now standard textbook material, we shall refer to the excellent texts (Hertz *et al* 1991; Haykin 1994; Bose & Liang 1996) for details, where references to the original works can be found.

The earliest instance of such an algorithm is perhaps the Widrow–Hoff algorithm (also known as the 'least mean square' algorithm (Haykin 1994, pp. 124–131)) which minimizes empirical risk w.r.t. the least squares criterion. So does the much publicised 'backpropagation algorithm' (Haykin, 1994, pp. 185–201) for training feedforward networks, which cleverly distributes the gradient computation into simple local computations at each node by exploiting the feedforward architecture and the chain rule of calculus. Other notable examples include the 'learning vector quantization' (LVQ) scheme of Kohonen for clustering data (Haykin 1994, pp. 427–430), which was analysed as such by Baras & La Vigna (1990) and Kosmotopoulos & Christodoulou (1996). Also fitting the bill are the algorithms for principal component analysis by Oja and Sanger (Hertz *et al*, 1991, pp. 199–209). The o.d.e. corresponding to the former was extensively analysed by (Wyatt & Elfadel 1995) and Leizarowitz (1997). Other examples can be found in the texts mentioned above.

We wish to point out here the immense potential 'two-timescale stochastic approximation' has for neural network training. Many neural network algorithms have two 'loops' where each iteration of the outer loop awaits completion (or near-completion) of the inner one – a situation tailor-made for the application of two-timescale stochastic approximation. The following (possibly incomplete) list of examples should make a convincing case.

(i) *The 'expectation-maximization' algorithm* – This usually refers to a scheme for maximizing the log-likelihood function under noisy or incomplete observations (Dempster *et al* 1977). We shall, however, interpret it in the broader sense where one aims to maximize or minimize an averaged quantity over a parameter and between two successive updates of the parameter (i.e., optimization steps), the expectation has to be recomputed under the current operative parameter. Barring some simple situations, the latter involves estimating the expectation by a time average based on Monte Carlo simulation. One can then do this averaging concurrently with the optimization recursions, but on a faster timescale to achieve the same effect. One potential application domain is the 'modular networks' of Haykin (1994, chap. 12). For another application to parameter optimization of hidden Markov models, see Bhatnagar & Borkar (1997b).

(ii) *Boltzmann machine* – The standard Boltzmann machine training algorithm (Haykin, 1994, pp. 318–330) requires estimation of certain correlations based on Monte Carlo simulation between two successive updates of the weights. Once again, the two can be made concurrent with the former being on a faster time scale.

(iii) *Training recurrent networks* – The 'recurrent back propagation' algorithm (Hertz *et al* 1991, pp. 172–176) alternate between a parameter update (based on the 'error' between desired outcomes and the actual outcomes of the relaxed network) and relaxation of the

network under current parameters. Again, these could be made concurrent if the relaxation takes place on a faster time scale.

(iv) *Learning time sequences* – Many algorithms for learning time sequences, such as 'real time recurrent learning' algorithm of Williams and Zipser (Hertz *et al* 1991, pp. 184–186) can be cast as two-time scale algorithms. See also Tsoi & Back (1994).

(v) *Competitive learning* – Many algorithms for unsupervised learning use competitive learning that involves two phases – the competition phase and the reward phase (Bose & Liang, 1996, pp. 344–351). The first selects a 'winner' by allowing some variation of a 'winner-take-all' network relax. The second 'rewards' the winner by performing a single parameter update that favours the winner. The two alternate. Clearly, this is another ideal set-up where two-time scale ideas could be used to make the two phases concurrent, the competition phase being faster.

4.5 *Reinforcement learning*

The term 'reinforcement learning' originates in studies of learning behaviour of animals. Mathematically, it falls somewhere in between the supervised and unsupervised learning paradigms of pattern recognition. Unlike supervised learning it does not call for exact information about 'error' from the environment. But unlike unsupervised learning which works with no information from the environment, reinforcement learning expects a 'reinforcement signal' from the environment indicating whether the recent move (i.e., parameter update) appears to be in the right direction or not. (Think of a 'critic' in place of a 'teacher'.) There are many variations of this theme, which have been very popular with the artificial intelligence community. Since the learning is 'incremental' (in the sense of making small adjustments to the parameter based on observations), and noisy, stochastic approximation is a popular paradigm for the same reasons as it was for economic applications. Extensive surveys appear in the articles by Sathiya Keerthi & Ravindran, (1994) and Kaelbling *et al* (1996) and in the books by Bertsekas & Tsitsiklis (1996) and Sutton & Barto (1998). We shall focus here only on the more recent applications to dynamic situations, i.e., to 'learning' control of Markov chains. See Puterman (1994) for a general background on controlled Markov chains.

Let $X_n, n = 0, 1, 2, \ldots,$ be a controlled Markov chain, governed by a control sequence $\{Z_n\}$ taking values in a finite action space A, with the dynamics described by:

$$P(X_{n+1} = j / X_m, Z_m, m \leq n) = p(X_n, j, Z_n), n \geq 0,$$

where $p : S \times S \times A \rightarrow [0, 1]$ is a prescribed 'transition probability function' satisfying

$$\sum_j p(i, j, a) = 1 \quad \forall i, a.$$

One aims to minimize over such $\{Z_n\}$ an appropriate cost, such as

(a) *discounted cost*

$$E\left[\sum_{n=0}^{\infty} \beta^n k(X_n, Z_n)\right], \quad \beta \in (0, 1),$$

(b) *average (or ergodic) cost*

$$\limsup_{n\to\infty} \frac{1}{n}\sum_{m=0}^{n-1} E[k(X_m, Z_m)],$$

where $k : S \times A \to R$ is a prescribed 'running cost' function.

For simplicity, we shall consider only the former. Define the 'value function' $V : S \to R$ by

$$V(x) = \inf E\left[\sum_{n=0}^{\infty} \beta^n k(X_n, Z_n)/X_0 = x\right], \quad x \in S.$$

Then $V(\cdot)$ is the unique solution to the dynamic programming equation

$$V(i) = \min_a (k(i,a) + \beta \sum_j p(i,j,a)V(j)), \quad i \in S.$$

Furthermore, if this minimum is attained at $a = v(i)$, then $Z_n = v(X_n), n \geq 0$, is an optimal choice. (In fact, the condition is both necessary and sufficient for such $\{Z_n\}$ to be optimal.) The issue then is to compute $V(\cdot)$. Two standard recursive schemes for this are:

Value iteration: Start with a guess V^0 and do:

$$V^{n+1}(i) = \min_a \left[k(i,a) + \beta \sum_j p(i,j,a)V^n(j)\right]$$

$n \geq 0$. Then $V^n \to V$.

(iii) *Policy iteration:* Start with $v_0 : S \to A$ and do for $n \geq 0$,

Step 1. Given $v_n(\cdot)$, find $V^n : S \to R$ satisfying $V^n(i) = k(i, v_n(i)) + \beta \sum_j p(i,j,v_n(i))V^n(j)$,

Step 2. Find $v_{n+1}(\cdot) \in \text{Argmin}\ (k(i,\cdot) + \beta \sum_j p(i,j,\cdot)V^n(j))$.

Then $V^n \to V$.

The two algorithms we sketch below, Q-learning algorithm and actor–critic algorithm, can be viewed, in some sense, as stochastic approximation counterparts of the above two schemes. The idea is as follows. These algorithms are used when the transition function $p(\cdot)$ is not explicitly available, but a transition with a prescribed probability $p(i,j,a)$ (say) can be 'simulated'. (Consider, e.g. a simulation of a large communication network where an exact dynamic model is intractable or computationally infeasible, but simulating the dynamics is easy based on 'local' dynamic rules.) One then wants to evaluate the conditional average w.r.t. $p(i,\cdot,a)$ through averaging. This can then be done by evaluating the function to be averaged at the destination of a simulated transition with this law and then 'seeing' the average by a stochastic approximation like dynamics.

This is only crude heuristics, because the algorithms are not exact analogs of the above, though they are derived from them. We start with Q-learning (Watkins 1989) where one works not with the value function, but the 'Q-value' defined by

$$Q(i,a) = k(i,a) + \beta \sum p(i,j,a)V(j), i \in S, a \in A.$$

Thus Q satisfies

$$Q(i,a) = k(i,a) + \beta \sum_j p(i,j,a) \min_b Q(j,b),$$

suggesting the iterative scheme

$$Q^{n+1}(i,a) = k(i,a) + \beta \sum_j p(i,j,a) \min_b Q^n(i,b),$$

if $p(\cdot)$ were available. In Q-learning, one replaces this by

$$Q^{(n+1)}(i,a) = (1 - a(n))Q^n(i,a) + a(n)\beta \min_b Q^n(\xi_n(i,a),b),$$

where $\xi_n(i,a)$ is a random variable with law $p(i,\cdot,a)$ independent of $\xi_n(j,b), Q^m(\cdot)$, $\xi_{m-1}(\cdot)$, $m \leq n, j \in S, b \in A, j \neq i, b \neq a$. This is the 'centralised' version of the algorithm. Most applications, however, call for a distributed, often asynchronous version. In particular, if only one component is updated at a time, that too corresponding to the current state-action pair of a controlled Markov chain, and in addition the control Z_n is obtained by minimizing or approximately minimizing $Q^n(X_n,\cdot)$, then this qualifies as a valid 'on line' adaptive control scheme. The general case has been analysed by Watkins & Dayan (1992), Tsitsiklis (1994) and Jaakola *et al* (1994).

Coming to the actor–critic algorithm, note that 'Step 1' of policy iteration could be replaced by a 'constant policy value iteration'

$$V^n_{m+1}(i) = k(i, v_n(i)) + \beta \sum_j p(i,j,v_n(i)) V^n_m(j),$$

for solving the linear system therein. This then becomes a two-loop algorithm. For its stochastic approximation counterpart, it is convenient to work with randomized policies $\varphi_n(\cdot)$ in place of $v_n(\cdot), \varphi_n : S \to \mathcal{P}(A) :=$ probability vectors indexed by A. Konda & Borkar (1997) propose two-timescale stochastic approximation algorithms where the value function is computed on a fast scale by an iteration akin to Q-learning and update $\varphi_n(\cdot)$ on a slower timescale by various reinforcement schemes. The algorithms differ only in the latter. They analyse asynchronous versions. The idea of such 'actor–critic' algorithms goes back to Barto *et al*, (1983). Many times the constant policy value iteration is 'naturally' very fast and the algorithm does well empirically even when the same time scale is employed for both components.

The counterparts of these for the average cost problem have been proposed and analysed by Abounady *et al* (1996b) and Konda & Borkar (1997) respectively. The methods of Borkar & Meyn (1997) take care of the stability issue for this class of algorithms.

It should, however, be kept in mind that these schemes suffer from the 'curse of dimensionality' for large problems. Trying to interface them with a suitable approximation scheme (e.g., by considering *a priori* a family of functions parameterized by a low dimensional parameter vector as candidates for $Q(\cdot)$ or $V(\cdot)$) is an active area of research (Tsitsiklis & Van Roy 1996, 1997; Marbach & Tsitsiklis 1998).

Finally, for an initial foray into reinforcement learning for continuous state space problems, see Baker (1997), Borkar (2000).

5. Related algorithms

In conclusion, we highlight some related classes of algorithms.

(i) *Stochastic automata algorithms* – These are algorithms for classification etc. which do fall in the general scheme above, but have sufficient special structure to merit separate classification. See Sastry & Thathachar (1999) for an extensive survey.

(ii) *Random search algorithms* – These algorithms sample the space through a random mechanism and then move so as to decrease the function one aims to minimize, at least 'on the average'. They may (usually do) combine a pure random search with features of gradient search, clustering algorithms etc. Examples are Matyas (1965), Solis & Wets (1981) and Boender *et al* 1982. Stochastic approximation often forms a component of these algorithms. For example, Santharam *et al* 1994 implement a Gaussian search mechanism where the mean and variance of the Gaussian are parameters tuned according to a stochastic approximation type scheme.

(iii) *Perturbation analysis and variants* – We have already mentioned these in § 4.1. We list them again here to emphasize the fact that they do not always fall in the stochastic approximation paradigm.

(iv) *Ordinal optimization* – The idea here is to do approximate optimization of the performance measure over a parameter by running several parallel simulations for different parameter values for a finite time and taking the best. The justification comes from the important observation that although the convergence of average performance to its equilibrium value may be slow for each parameter value, their rank ordering occurs relatively fast. See Ho *et al* (1992) for early work in this direction and Ho (1994) for an interesting discussion. Dai (1995) and Xie (1995) provide theoretical analysis of this scheme.

(v) *Simulated annealing* – This is a stochastic algorithm for global optimization, usually studied in the discrete time, discrete domain set up. Suppose we want to minimize a function $f(\cdot)$ on a discrete (say, finite) set D. If we can generate a random variable X with distribution

$$P(X = i) = Ce^{-f(i)/T} := \pi_T(i),$$

where C is the normalizing factor, it will have high probability of being at the global minimum. In fact, the lower the parameter T (called 'temperature' by analogy with physics), the higher this probability. The Metropolis algorithm (Metropolis *et al* 1953) generates a D-valued Markov chain with stationary probability $\pi_T(\cdot)$. The simulated annealing algorithm runs a similar time-inhomogeneous chain where the parameter T is decreased very slowly so as to track $\pi_T(\cdot)$ as $T \to 0$, concentrating on global minima in the limit. The reason for using this ploy instead of directly generating random variables with distribution $\{\pi_T\}$ is that the transition probability for these chains depends only on the differences $f(i) - f(j)$, between 'neighbouring' i, j and not on $f(i), f(j)$ separately and also because in typical applications of this algorithm (combinatorial optimization, image processing, ...), the former is much easier to calculate than the latter. See Aarts & Korst (1989) for an overview and Desai (1999) for some theoretical aspects.

The continuous state space analog leads to the Gelfand–Mitter algorithm we encountered in § 3.2. It is instructive to view (1) with $h(\cdot) = -\nabla f$ as being somewhere between this and the deterministic gradient scheme, i.e., as 'suboptimally cooled' simulated annealing algorithm.

(vi) *Correlation-based methods* – Note that for $1 \leq i \leq d$, $(f(X(n+1)) - f(X(n)) \cdot (X_i(n+1) - X_i(n))$ has the same sign as $(f(X(n+1)) - f(X(n))/(X_i(n+1) - X_i(n))$ whenever the denominator of the latter is nonzero. Since the latter ratio can be viewed as a finite difference approximation to $(\partial f/\partial x_i)(X(n))$, the vector whose ith component is $(f(X(n+1)) - f(X(n))) \cdot (X_i(n+1) - X_i(n))$ is an approximate descent direction. This is the intuition behind correlation-based optimization schemes. Using 'one-step correlation' as above, however, leads to a numerically ill-behaved algorithm, so one replaces it by the empirical correlations

$$\frac{1}{M}\sum_{m=1}^{M}(f(X(n-m)) - f(X(n-m-N))) \cdot (X_i(n-m) - X_i(n-m-N)),$$

for $1 \leq i \leq d$, with $M, N \geq 1$ being prescribed integers. Bharath & Borkar (1998) describe such a scheme with the additional feature that it is interfaced with Fabian's 'sign' algorithm, i.e., one uses the componentwise sign of the empirical correlations rather than the empirical correlations themselves. They use a two-time scale scheme to achieve the averaging.

An older correlation based scheme is the Alopex algorithm of Harth & Tzanakou (1974) which uses a similar (i.e., ± 1-valued) vector as a driving term of the algorithm, obtained from the empirical correlations. The difference is that it is not merely the sign thereof, but is generated by a Gibbsian mechanism. See Sastry & Unnikrishnan (1997) for a theoretical analysis of this algorithm.

(vii) *Continuous time algorithms* – For continuous time analogs of stochastic approximation, Mel'nikov (1996) provides an extensive survey and bibliography.

The work of the second author is supported by the DST grant III 5(12)/96-ET.

References

Aarts E, Korst J 1989 *Simulated annealing and Boltzmann machines: A stochastic approach to combinatorial optimization and neural computing* (New York: John Wiley)

Abdelhamid S 1973 Transformations of observations in stochastic approximation. *Ann. Stat.* 1: 1158–1174

Abounady J, Bertsekas D, Borkar V 1996a Stochastic approximation for nonexpansive maps: application to Q-learning algorithms Preprint no. P-2433, Laboratory for Information and Decision Sciences, MIT, Cambridge, MA

Abounady J, Bertsekas D, Borkar V 1996b Learning algorithms for Markov decision processes with average cost Preprint no. P-2434, Laboratory for Information and Decision Sciences, MIT, Cambridge, MA

Anbar D 1973 On optimal estimation methods using stochastic approximation procedures. *Ann. Stat.* 1: 1175–1184

Arthur B 1994 *Increasing returns and path dependence in the economy* (Ann Arbor: Uni. of Michigan Press)

Arthur B, Ermoliev Y, Kaniovski Y 1983 A generalized urn problem and its applications. *Cybernetics* 19: 61–71

Baker W 1997 *Learning via stochastic approximation in function space*. PhD thesis, Harvard University, Cambridge, MA

Baras J, LaVigna A 1990 Convergence of Kohonen's learning vector quantization. *Proc. Int. Joint Conf. Neural Networks* (San Diego, CA) vol. 3

Barto A, Sutton R, Anderson C 1983 Neuron-like elements that can solve difficult learning control problems. *IEEE Trans. Syst., Man Cybern.* 13: 835–846

Basak G, Hu I, Wey C-Z 1997 Weak convergence of recursions. *Stoch. Processes Appl.* 68: 65–82

Benaim M 1996 A dynamical system approach to stochastic optimization. *SIAM J. Control Optim.* 34: 437–472

Benaim M 1998 Recursive algorithms, urn processes and chaining number of chain recurrent sets. *Ergodic Theory Dyn. Syst.* 18: 53–87

Benaim M 1998 Dynamics of stochastic approximation algorithms *Le Seminaire de Probabilites* (to appear)

Benaim M, Hirsch M 1997 Stochastic adaptive behaviour for prisoner's dilemma (preprint)

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

Bertsekas D, Tsitsiklis J 1989 *Parallel and distributed computation: Numerical methods* (Englewood Cliffs, NJ: Prentice Hall)

Bertsekas D, Tsitsiklis J 1997 *Neurodynamic programming* (Belmont, MA: Athena Scientific)

Bertsekas D. Tsitsiklis J 1999 Gradient convergence in gradient methods. *SIAM J. Control Optim.* (to appear)

Bhatnagar S, Borkar V 1997 Multiscale stochastic approximation for parametric optimization of hidden Markov models. *Probab. Eng. Inf. Sci.* 11: 509–522

Bhatnagar S, Borkar V 1998 A two time-scale stochastic approximation scheme for simulation-based parametric optimization. *Probab. Eng. Inf. Sci.* 12: 519–531

Bharath B, Borkar V 1998 Robust parameter optimization of hidden Markov models. to appear in *J. Indian Inst. Sci.* 78: 119–130

Boender C, Rinnooy Kan A, Timmer G, Stougie L 1982 A stochastic method for global optimization. *Math. Program.* 22: 125–140

Borgers T, Sarin R 1997 Learning through reinforcement and replicator dynamics. *J. Econ. Theory* 77: 1–14

Borkar V 1995 *Probability theory: An advanced course* (New York: Springer Verlag)

Borkar V 1996a Recursive self-tuning control of finite Markov chains. *Appl. Math.* 24: 169–188

Borkar V 1996b Distributed computation of fixed points of ∞-nonexpansive maps. *Indian Acad. Sci. Proc. Math. Sci.* 106: 289–300

Borkar V 1996c Stochastic approximation with two time scales. *Syst. Control Lett.* 29: 291–294

Borkar V 1998 Asynchronous stochastic approximation.*SIAM J. Control Optim.* 36: 840–851

Borkar V 2000 A learning algorithm for discrete time stochastic control. *Probability in Engineering and Informational Sciences* (to appear)

Borkar V, Meyn S 1999 The O.D.E. method for convergence of stochastic approximation and reinforcement learning algorithms. *SIAM J. Control Optim.* (to appear)

Borkar V, Mitter S 1999 A strong approximation theorem for stochastic recursive algorithms. *J. Optim. Theor. Appl.* 10: 499–513

Bose N, Liang P 1996 *Neural networks fundamentals with graphs, algorithms and applications* (New York: McGraw Hill)

Brandiere O 1998a Some pathological traps for stochastic approximation *SIAM J. Control Optim.* 36: 1293–1314

Brandiere O 1998b The dynamic system method and the traps *Adv. Appl. Probab.* 30: 137–151

Brandiere O, Duflo M 1996 Les algorithms stochastiques contournent – ils les pieges? *Ann. Inst. Henri Poincare* 32: 395–427

Brown G 1951 Iterative solutions of games by fictitious play. In *Activity analysis of production and allocation* (ed.) T Koopmans (New York: John Wiley)

Chazan D, Miranker W 1969 Chaotic oscillations. *Linear Algebra Appl.* 2: 199–222

Chen H 1994 Stochastic approximation and its new applications. *Proc. 1994 Hong Kong Int. Workshop on New Directions in Control and Manufacturing*, pp 2–12

Chen H 1998a Convergence rate of stochastic approximation algorithm in the degenerate case. *SIAM J. Control Optim.* 36: 100–114

Chen H 1998b Convergence of SA algorithms in multi-root or multi-extreme cases. *Stoch. Stoch. Rep.* 64: 255–266

Chen H, Guo L 1991 *Identification and stochastic adaptive control* (Boston: Birkhauser)

Chong E, Ramadge P 1994 Stochastic optimization of regenerative systems using infinitesimal perturbation analysis. *IEEE Trans. Autom. Control* 39: 1400–1410

Dai L 1995 Convergence properties of ordinal comparison in the simulation of discrete event dynamic systems. *Proc. 34th IEEE Conf. on Decision and Control* pp. 2604–2609

Delyon B 1996 General results on the convergence of stochastic algorithms. *IEEE Trans. Autom. Control* 41: 1245–1255

Dempster A, Laird N, Rubin D 1977 Maximum likelihood from incomplete data via the EM algorithm. (with discussion). *J. R. Stat. Soc. Series* B39: 1–38

Desai M 1998 Some recent results characterizing the finite time behaviour of the simulated annealing algorithm. *Sadhana* 24: 317–337

Dippon J, Renz J 1997 Weighted means in stochastic approximation of minima. *SIAM J. Control Optim.* 35: 1811–1827

Duflo M 1997 *Random iterative models* (Berlin-Heidelberg: Springer)

Dupuis P 1988 Large derivations analysis of some recursive algorithms with state-dependent noise. *Ann. Probab.* 16: 1509–1536

Dupuis P, Kushner H 1985 Stochastic approximation via large derivations: asymptotic properties. *SIAM J. Control Optim.* 23: 675–696

Dupuis P, Kushner H 1989 Stochastic approximation and large derivations: upper bounds and w.p. 1 convergence. *SIAM J. Control Optim.* 27: 1108–1135

El Fattah Y 1981 Gradient approach for recursive estimation and control in finite Markov chains. *Adv. Appl. Probab.* 13: 778–803

Fabian V 1960 Stochastic approximation methods. *Czech. Math. J.* 10

Fang H, Gong G, Quian M 1997 Annealing of iterative stochastic schemes. *SIAM J. Control Optim.* 35: 1886–1907

Fogel E 1981 A fundamental approach to convergence analysis of least squares algorithms. *IEEE Trans. Autom. Control* 26: 646–655

Freidlin M, Wentzell A 1984 *Random perturbations of dynamical systems* (Berlin – New York: Springer Verlag)

Fudenberg D, Kreps D 1993 Learning mixed equilibria. *Games Econ. Behav.* 5: 320–367

Fudenberg D, Levine D 1998 Theory of learning in games (Cambridge, Mass: MIT Press)

Gardner W 1984 Learning characteristics of stochastic-gradient-descent algorithms: a general study, analysis, and critique. *Signal Process.* 6: 113–133

Gelfand S, Mitter S 1991 Recursive stochastic algorithms for global optimization in R^d. *SIAM J. Control Optim.* 29: 999–1018

Gharavi R, Anantharam V 1999 Structure theorems for partially asynchronous iterations of a nonnegative matrix with random delays. *Sadhana* 24: 369–423

Gidas B 1985 Global minimization via the Langevin equation. *Proc. 24th Conf. on Decision and Control* Ft. Lauderdale, pp. 774–778

Gong W, Ho Y-C 1987 Smoothed conditional perturbation analysis of discrete event dynamic systems. *IEEE Trans. Autom. Control* 32: 858–866

Hardle W, Nixdorf R 1987 Nonparametric sequential estimation of zeros and extrema of regression functions. *IEEE Trans. Inf. Theory* 33: 367–372

Harsanyi J, Selten R 1988 *General theory of equilibrium selection in games* (Cambridge, MA: MIT Press)

Harth E, Tzanakou E 1974 Alopex: a stochastic method for determining visual receptive fields. *Vision Res.* 14: 1475–1482

Haykin S 1991 *Adaptive filter theory* 2nd edn. (Englewood Cliffs, NJ: Prentice Hall)

Haykin S 1994 *Neural networks: A comprehensive foundation* (New York: McMillan)

Hertz J, Krogh A, Palmer R 1991 *An introduction to the theory of neural computation* (Redwood City, CA: Addison Wesley)

Hirsch M 1989 Convergent activation dynamics in continuous time networks. *Neural Networks* 2: 331–349

Hofbauer J, Siegmund K 1988 *The theory of evolution and dynamical systems* (Cambridge: University Press)

Ho Y-C 1994 Heuristics, rules of thumb and the 80/20 proposition. *IEEE Trans. Autom. Control* 39: 1025–1027

Ho Y-C, Cao X 1991 *Perturbation analysis of discrete event dynamical systems* (Boston: Birkhauser)

Ho Y-C, Cao X, Cassandras C 1983 Infinitesimal and finite perturbation analysis for queuing networks. *Automatica* 19: 419–445

Ho Y-C, Li S 1988 Extensions of infinitesimal perturbation analysis. *IEEE Trans. Autom. Control* 33: 427–438

Ho Y-C, Srinivas R, Vakili P 1992 Ordinal optimization of discrete event dynamic systems. *J. Discrete Events Dyn. Syst.* 2: 61–88

Jaakola T, Jordan M, Singh S 1994 On the convergence of stochastic iterative dynamic programming algorithms. *Neural Comput.* 6: 1185–1201

Kaelbling L, Littman M, Moore A 1996 Reinforcement learning: a survey. *J. Artif. Intell. Res.* 4: 237–285

Kaniovski Y, Young H P 1995 Learning dynamics in games with stochastic perturbations. *Games Econ. Behav.* 11: 330–363

Katkovnik V, Kulchitsky Y 1972 Convergence of a class of random search algorithms. *Autom. Remote Control* 8: 1321–1326

Kiefer J, Wolfowitz J 1952 Stochastic estimation of the maximum of a regression function. *Ann. Math. Stat.* 23: 462–466

Konda V, Borkar V 1999 Actor-critic type learning algorithms for Markov decision processes. *SIAM J. Control Optim.* (to appear)

Kosmotopoulos E, Christodoulou M 1996 Convergence properties of a class of learning vector quantization algorithms. *IEEE Trans. Image Process.* 5: 361–368

Kulkarni S, Horn C 1996 An alternative proof for convergence of stochastic approximation algorithms. *IEEE Trans. Autom Control* 41: 419–424

Kushner H 1984 *Approximation and weak convergence methods for random processes with application to stochastic systems theory* (Cambridge, MA: MIT Press)

Kushner H, Clark D 1978 *Stochastic approximation for constrained and unconstrained systems* (New York: Springer Verlag)

Kushner H, Yang J 1995 Stochastic approximation with averaging and feedback: rapidly convergent "on-line" algorithm. *IEEE Trans. Autom. Control* 40: 24–34

Kushner H, Yin G 1987a Asymptotic properties for distributed and communicating stochastic approximation algorithm. *SIAM J. Control Optim.*25: 1266–1290

Kushner H, Yin G 1987b Stochastic approximation algorithms for parallel and distributed processing. *Stoch. Stoch. Rep.* 22: 219–250

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

Leizarowitz A 1997 Convergence of solutions to equations arising in neural networks. *J. Optim. Theory Appl.* 94: 533–560

Ljung L 1977a On positive real transfer functions and the convergence of some recursive schemes. *IEEE Trans. Autom. Control* 22: 539–550

Ljung L 1977b Analysis of recursive stochastic algorithms. *IEEE Trans. Autom. Control* 22: 551–575

Ljung L 1978 Strong convergence of a stochastic approximation algorithm. *Ann. Stat.* 6: 680–696

Ljung L 1986 *System identification: Theory for the user* (Englewood Cliffs, NJ: Prentice Hall)

Ljung L, Soderstrom T 1983 *Theory and practice of recursive identification* (Cambridge, MA: MIT Press)

Ljung L, Pflug G, Walk H 1992 *Stochastic approximation and optimization of random systems* (Basel: Birkhauser)

Marbach P, Tsitsiklis J. 1998 Simulation-based optimization of Markov reward processes (preprint)
Matyas J 1965 Random optimization. *Autom. Remote Control (Engl. Transl.)* 26: 246–253
Mel'nikov A 1996 Stochastc differential equations: singularity of coefficents, regression models, and stochastic approximation. *Russ. Math. Surv.* 51: 819–909
Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E 1953 Equations of state calculations by fast computing machines. *J. Chem. Phys.* 21: 1087–1091
Meyn S, Tweedie R 1994 *Markov chains and stochastic stability* (New York: Springer Verlag)
Nazin A, Polyak B, Tsybakov A 1990 Passive stochastic approximation. *Autom. Remote Control (Engl. Transl.)* 50: 1563–1569
Nevelson M, Khasminskii R 1976 *Stochastic approximation and recursive estimation.* AMS Trans. Math. Monographs 47 (Providence, RI: Am. Math. Soc.)
Pelletier M 1998 On the almost sure asymptotic behaviour of stochastic algorithms *Stoch. Processes Appl.* 78: 217–244
Pemantle R 1990 Nonconvergence to unstable points in urn models and stochastic approximations. *Ann. Probab.* 18: 698–712
Pflug G 1986 Stochastic optimization with constant stepsize: Asymptotic laws. *SIAM J. Control Optim.* 24: 655–666
Pflug 1996 Searching for the best: stochastic approximation, simulated annealing and related procedures. *Identification, adaptation, learning* (eds) S Bittanti and G Picci (Berlin: Springer Verlag) pp. 514–549
Polyak B 1991 New stochastic approximation type procedures. *Autom. Remote Control (Engl. Trans.)* 51: 937–946
Polyak B, Juditsky A 1992 Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.* 30: 838–855
Polyak B, Tsypkin Y 1973 Pseudogradient adaptation and training algorithms. *Autom. Remote Control* 34: 377–397
Puterman M 1994 *Markov decision processes* (New York: John Wiley)
Rachev S, Ruschendorf L 1995 Probability metrics and recursive algorithms. *Adv. Appl. Probab.* 27: 770–799
Ripley B 1987 *Stochastic simulation* (New York: John Wiley)
Robbins H, Monro 1951 A stochastic approximation method. *Ann. Math. Stat.* 22: 400–407
Rubinstein R 1981 *Simulation and the Monte Carlo method* (New York: John Wiley)
Ruppert D 1985 A Newton–Raphson version of the multivariate Robbins–Monro procedure. *Ann. Stat.* 13: 236–245
Ruppert D 1988 Efficient estimators for a slowly convergent Robbins-Monro process. Report 781, *School of OR and IE Tech. Cornell University*, NY
Ruszczynski A, Syski W 1983 Stochastic approximation method with gradient averaging for unconstrained problems. *IEEE Trans. Autom. Control* 28: 1097–1105
Santharam G, Sastry P S, Thathachar M 1994 Continuous action set learning automata for stochastic optimization. *J. Franklin Inst.* B331: 607–628
Sargent T 1993 *Bounded Rationality in Macroeconomics* (Oxford: Clarendon Press)
Sastry P S, Thathachar M 1999 Learning automata algorithms for pattern classification. *Sadhana* 24: 261–292
Sastry P S, Unnikrishnan K 1997 Analysis of Alopex optimization algorithm (preprint)
Sathiya Keerthi S, Ravindran B 1994 A tutorial survey of reinforcement learning. *Sadhana* 19: 851–889
Solis F, Wets R 1981 Minimization by random search techniques. *Math. OR* 6: 19–30
Solo V, Kong X 1995 *Adaptive signal processing algorithms: Stability and performance* (Englewood Cliffs, NJ: Prentice Hall)
Spall J 1992 Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control* 37: 332–341
Stratonovich R 1968 Does there exist a theory of optimal, adaptive and self-adaptive systems? *Autom. Remote Control (Engl. Trans.)* 29: 83–93

Sutton R, Barto A. 1998 *Reinforcement learning* (Cambridge, Mass.: MIT Press)
Tadic V 1998 Stochastic approximation with random truncations, state-dependent noise and discontinuous dynamics. *Stoch. Stoch. Rep.* 64: 283–326
Thathachar M. Sastry P 1987 Learning optimal discriminant functions through a cooperative game of automata. *IEEE Trans. Syst., Man and Cybern* 17: 73–85
Tsitisklis J 1994 Asynchronous stochastic approximation and Q-learning. *Mach. Learning* 16: 185–202
Tsitisklis J, Bertsekas D, Athans M 1986 Distributed asynchronous deterministic and stochastic gradient optimization algorithms.*IEEE Trans. Autom. Control* 31: 803–812
Tsitsiklis J, Van Roy B 1996 Feature-based methods for large scale dynamic programming *Mach. Learning* 22: 59–94
Tsitsiklis J, Van Roy B 1997 An analysis of temporal-difference learning with function approximation. *IEEE Trans. Autom. Control* 42: 674–690
Tsoi A, Back A 1994 Locally recurrent globally feedforward networks: a critical review of architectures. *IEEE Trans. Neural Networks* 5: 229–239
Tsypkin Y 1966 Adaptation, learning and self-learning in automatic systems. *Autom. Remote Control (Engl. Trans.)* 27: 16–51
Tsypkin Y 1968 All the same, does a theory of synthesis of optimal adaptive systems exist ? *Autom. Remote Control (Engl. Trans.)* 29: 93–99
Tsypkin Y 1971 *Adaptation and learning in automatic systems* (New York: Academic Press)
Vapnik V 1995 *The nature of statistical learning theory* (New York: Springer Verlag)
Vega-Redondo F 1996 *Evolution, games and economic behaviour* (Oxford: University Press)
Wang I, Chong E, Kulkarni S 1996 Equivalent necessary and sufficient conditions on noise sequence for stochastic approximation (preprint)
Wasan M 1969 *Stochastic approximation* (Cambridge: University Press)
Watkins C 1989 *Learning from delayed rewards.* PhD thesis, Cambridge University
Watkins C, Dayan P 1992 Q-learning. *Mach. Learning* 8: 279–292
Widrow B, Stearns S 1985 *Adaptive signal processing* (Englewood Cliffs, NJ: Prentice Hall)
Wilson F 1969 Smoothing derivatives of functions and applications. *Trans. Am. Math. Soc.* 139: 413–428
Wyatt J, Elfadel I 1995 Time-domain solutions of Oja's equations. *Neural Comput.* 7: 915–922
Xie X 1995 Dynamics and convergence rate of ordinal comparison of stochastic discrete event systems (preprint)
Yin G 1981 On extensions of Polyak's averaging approach to stochastic approximation. *Stoch. Stoch Rep.* 36: 245–264
Yin G, Yin K 1994 Asymptotically optimal rate of convergence for smoothed stochastic recursive algorithms. *Stoch. Stoch. Rep.* 47: 21–46
Yin G, Yin K 1996 Passive stochastic approximation with constant step size and window width. *IEEE Trans. Autom. Control* 41: 90–106
Young H P 1998 *Individual strategy and social structure* (Princeton, NJ: University Press)