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