Journal of Computational Mathematics, Vol.12, No.1, 1994, 21–30.

STOCHASTIC APPROXIMATION IN REAL TIME: A PIPE LINE APPROACH*

Zhu Yun-min[1)]
(*Inst. of Math. Scis., Chengdu Branch, Academia Sinica, Chengdu, Sichuan, China*)
Yin Gang[2)]
(*Department of Mathematics, Wayne State University, Detroit, USA.*)

Abstract

A new approach for stochastic approximation in real time is developed. A number of processors are simultaneously active to carry out a computing task. All processors work on the same system with different starting time. After each iteration, computed data are passed to the next processor on line. Interacting tasks and iterative instructions are carried through pipelining of computation and communication. Asymptotic properties of the algorithm are developed, and comparisons of the performance between the new algorithm and the classical one are made.

1. Introduction

The objective of this work is to study stochastic approximation in real time. A pipe line approach is suggested. Asymptotic properties of the procedure are developed, and comparisons of rate of convergence with the classical algorithms are made.

Let $x \in \mathbf{R}^L$, and $f(\cdot) : \mathbf{R}^L \mapsto \mathbf{R}^L$. The traditional stochastic approximation methods deal with the problem of finding the roots of $f(x) = 0$ by using noisy measurements $Y_{\hat{n}}$ via the recursive procedure

$$X_{\hat{n}+1} = X_{\hat{n}} + a_{\hat{n}} Y_{\hat{n}},\ Y_{\hat{n}} = f(X_{\hat{n}}) + \xi_{\hat{n}} \tag{1.1}$$

where the gains $a_{\hat{n}}$ satisfy $a_{\hat{n}} > 0$, $\sum a_{\hat{n}} = \infty$, $\frac{a_{\hat{n}} - a_{\hat{n}+1}}{a_{\hat{n}}} \xrightarrow{\hat{n}} 0$, and $\xi_{\hat{n}}$ represent the measurement errors. Various successful applications of the stochastic approximation methods have been reported[1,2].

The Robbins-Monro (RM) algorithm (1.1) can be thought of as a two phase operation. For each iteration, the first phase is to take measurement $Y_{\hat{n}}$, and the next step is to form the new estimate $X_{\hat{n}+1}$ by means of addition. Usually, most of the computation time is spent on the process of collecting data $Y_{\hat{n}}$. The second step $X_{\hat{n}} + a_{\hat{n}} Y_{\hat{n}}$ is less

* Received May 21, 1991.
1) The Project Supported in part by Science Foundation of Academia Sinica.
2) The Project Supported by National Natural Science Foundation of China, and by Wayne State University.

time consuming. This feature is, however, not reflected in (1.1) since $\hat{n}$ represents the iteration number rather than real time.

In recent years, parallel processing methods have drawn much of attention. Starting from [3], several stochastic approximation algorithms have been developed for parallel and distributed computing[4–6]. An extensive survey can be found in [7]. In view of the recent developments, and motivated by the idea of pipelining of computation for large scale parallelization[8], a new algorithm for stochastic approximation in real time is suggested here. In lieu of using a single processor alone as in the traditional setting, a number of identical processors are utilized to update the same system. The processors are lined up as on a production line. After one step iteration is completed, the newly computed data are passed to the next processor on the line. Each processor, repeatedly, executes the same instruction on successive observed data and data received from the preceding processor.

Assume that a single processor needs r units of time for a phase one operation and 1 unit of time for an addition; let n denote real time. Instead of algorithm (1.1), $r+1$ processors will be used to carry out the computing task. All $r+1$ processors work on the same system vector X, and communicate with each other through pipe line structure. In spite of different time indices, the scheme is the same for all the processors. Thus, it suffices to use a single formula to describe the procedure. The initial conditions are given for $X_1, X_2, \cdot, X_{r+2}$. The observation at time n is Y_n. The algorithm is given by

$$X_{n+1} = X_n + a_{n-r} Y_{n-r} \tag{1.2}$$

where the gains a_n are as before with $\hat{n}$ replaced by n. When $r = 0$, the above algorithm formally reduces to the classical stochastic approximation procedure.

In the proposed algorithm, the overall system consists of a number of parallel processors connected through communications. The length of a computation cycle is equal to the number of processors participating in the computation, which is also equal to the time required for a single processor to complete one iteration. The notation $\{Y_{n-r}\}$ means that the measurement was begun r (time) units before, and completed at time n. We shall call such measurements "delayed" measurements. In fact, in various situations, the observed signals are rarely available immediately without any delays. For example, for the Viterbi decoding algorithm, the desired signals are not available until several symbol intervals later. Therefore, even for a single processor alone, an algorithm with delay seems to be a more natural model.

The remainder of this paper is arranged as follows. A modified algorithm is given first, and the strong convergence is obtained. An order of magnitude estimate of the algorithm is derived in Section 3, and asymptotic normality is proved in Section 4. Further discussions and comparisons of rates of convergence are given in Section 5. Finally, an appendix is included. In the sequel, "$\prime$" stands for the transpose of a matrix; $g_x(\cdot)$ denotes the gradient of a function $g(\cdot)$ and K denotes a generic positive constant with possibly different values.