Signal Processing 6 (1984) 113–133
North-Holland

LEARNING CHARACTERISTICS OF STOCHASTIC-GRADIENT-DESCENT ALGORITHMS: A GENERAL STUDY, ANALYSIS, AND CRITIQUE

W.A. GARDNER

Signal and Image Processing Laboratory, Department of Electrical and Computer Engineering, University of California, Davis 95616, USA

Received 13th June 1983

Abstract. A comprehensive analysis of the mean-square learning characteristics of stochastic-gradient-descent algorithms is presented. The approach is based on the commonly exploited simplifying assumption of stationary independent training vectors. Characteristics analyzed include stability, steady-state misadjustment, initial rate of convergence, optimum step size, and steady-state autocovariance and spectral characteristics of the weight-vector. Effects on these characteristics due to degree of randomness of stochastic gradient, particular data distribution, and data corruption are isolated and analyzed. An objective of the work is to keep the number of simplifying assumptions and approximations to a minimum. Comparisons of results with previous more approximate analyses are made.

Zusammenfassung. Lernkurven von stochastischen Gradienten Algorithmen werden untersucht. Die vereinfachende Annahme von stationären, unabhängigen Trainingsvektoren wird benutzt. Charakteristiken die untersucht werden beinhalten Stabilität, 'steady state' Fehlanpassung, Start Konvergenz, optimaler Schritt sowie 'steady state' Autokovarianz und spektrale Charakteristik des Gewichtsvektors. Die Effekten auf diese Charakteristiken von dem Zufälligkeitsgrad des stochastischen Gradienten, besonderer Daten Verteilung und Daten Verderbung werden isoliert und analysiert. Ein Zielpunkt dieser Arbeit ist es die Anzahl vereinfachenden Annahmen und Approximationen auf ein minimum zu beschränken. Vergleiche mit Resultaten die von approximativeren Analysen stammen werden gezogen.

Résumé. Une analyse d'ensemble des caractéristiques de convergence de l'erreur quadratique moyenne dans les algorithmes utilisant la décroissance du gradient stochastique est presentée. Cette approche est basée sur l'hypothése simplificatrice classique de stationnarité et indépendance des vecteurs de test. Les caractéristiques analysées comprennent la stabilité, l'écart d'état stable, la vitesse initiale de convergence, le pas optimum d'incrémentation, et l'autocovariance de l'état stable, ainsi que les caractéristiques spectrales du vecteur de pondération. Les effets sur ces caractéristiques de la variance du gradient stochastique, avec en particulier la distribution des données et leur dégradation, sont isolées et analysées. Un des objectifs de ce travail est de garder le nombre d'hypothéses simplificatrices et approximations minimal. Les résultats obtenus sont comparés aux analyses antérieures basées sur de plus larges approximations.

1. Introduction

1.A. Purpose and overview

The LMS algorithm for adaptive linear estimation has been used in a wide variety of applications including sensor array processing, channel equalization, echo and other noise and interference cancellation, and system identification. The most attractive feature of the LMS algorithm is its simplicity and corresponding amenability to simple implementation. The most detractive feature, on the other hand, is its relatively slow convergence (e.g., by comparison with recursive least squares and Kalman algorithms [1, and references therein]). Consequently the LMS algorithm is viable—and in fact popular—in applications where simplicity is important, and requirements on speed of convergence are not too stringent.

Although the behavior of the LMS algorithm has been studied at length by many, and is fairly well understood in general, there remains a need for a relatively comprehensive analysis that explains the effects on analytical results of many

of the commonly used simplifying assumptions and approximations such as non-random data (or, equivalently, assuming the weight vector variance is negligible by focusing on the mean exclusively), Gaussian random data (or, equivalently, assuming that higher than second moments can be deleted or approximated), uncorrupted training signals, and small step-size. In order for such a relatively comprehensive analysis to be tractable, there is one simplifying assumption that cannot be removed, and therefore whose effects on analytical results cannot be determined (analytically). This is the commonly exploited assumption that the training data is a sequence of statistically independent random vectors. Nevertheless, it has been experimentally verified by many that when the step size is sufficiently small (and the convergence therefore sufficiently slow) analyses of algorithm behavior based on this independence assumption agree closely with empirical evaluations of algorithm behavior (e.g., [2]–[9]). Moreover, recent research has provided analytical verification of the fact that the independence assumption will yield accurate analyses if the step size is sufficiently small [10]–[13]. To supplement this recent progress, this paper carries through exact analyses without invoking the commonly used simplifying assumptions and approximations—except for the independence assumption—to obtain quantitative evaluations of the accuracy of previous approximate analyses.

Unfortunately, the relatively comprehensive analysis presented herein, based on the independence assumption, coupled with the recent analysis of the independence assumption itself, still does not provide a complete self-consistent analysis, since the independence assumption apparently cannot be analytically justified for relatively large step sizes (which is, indeed, a case of practical interest, as explained in Section 4.C). But this is perhaps the best that can be done from the pragmatic point of view of obtaining a good tradeoff between model realism and model tractability. Consequently, analyses of algorithm behavior must continue to be verified, augmented, and in some cases invalidated by empirical evaluations and/or simulations. Nevertheless, model tractability and its necessary complement, empirical evaluation, are essential components of the scientific method.

In Section 1.B, the linear mean-square estimation problem, and the stochastic-gradient-descent approach to solution are formulated. Then a variety of applications are briefly described. The section concludes with a description of the simplifying assumptions exploited in the analysis presented in Sections 2 and 3.

In Section 2, several fundamental learning characteristics are defined. These are stability, misadjustment, initial rate of convergence, and optimum step size (the important time-to-convergence characteristic is not analyzed herein, for reasons explained in Section 4.F). Then effects on learning characteristics, of several fundamental features of the algorithm and data, are isolated and analyzed. These features include randomness of data, degree of randomness of stochastic gradient, the particular probability distribution of the data, and data corruption.

In Section 3, general formulas for learning characteristics are derived. These include bounds and approximations for arbitrary data distribution in subsections 3.A and 3.D, and exact formulas for the case of Gaussian data in subsection 3.C. Also, an exact, linear, time-invariant recursion for the learning curve for arbitrary data distribution is derived in subsection 3.B.

In Section 4 a critique of previous closely related studies is given. Specific comparisons of results and clarifications of simplifying assumptions and approximations are included.

1.B. Problem formulation

Consider the problem of adaptive estimation of a desired possibly random sequence $\{d(i)\}$ with the random sequence of linear estimators[1] $\{\hat{d}(i)\}$,

$$\hat{d}(i) = \sum_{n=1}^{N} w_n(i)x_n(i) = \boldsymbol{W}^{\mathrm{T}}(i)\boldsymbol{X}(i), \qquad (1)$$

[1] For a matrix $\boldsymbol{M}$, $\boldsymbol{M}^{\mathrm{T}}$ denotes the transpose of $\boldsymbol{M}$, and for a square matrix, $\mathrm{tr}\{\boldsymbol{M}\}$ denotes the trace of $\boldsymbol{M}$, and $\boldsymbol{M}^2 = \boldsymbol{M}\boldsymbol{M}$ denotes the product of $\boldsymbol{M}$ with itself.

where $\boldsymbol{W}(i)$ is the estimation weight-vector, and $\boldsymbol{X}(i)$ is the observed data vector. This paper is concerned with the transient and steady-state behavior of the sequence of weight-vectors $\{\boldsymbol{W}(i)\}$ and corresponding estimators $\{\hat{d}(i)\}$, when the following stochastic-gradient-descent (SGD) algorithm is used for adaptation:

$$\boldsymbol{W}([i+1]K) = \boldsymbol{W}(iK) - \frac{\mu}{2}\hat{\nabla}\varepsilon(iK),$$

$$\hat{\nabla}\varepsilon(iK) = -\frac{2}{K}\sum_{q=0}^{K-1} e(iK+q)\boldsymbol{X}(iK+q), \quad (2)$$

$$\boldsymbol{W}(iK+q) = \boldsymbol{W}(iK), \quad q = 1, 2, \ldots, K-1.$$

In this algorithm, $\hat{\nabla}\varepsilon$ is the stochastic gradient of the time-averaged squared stochastic error

$$\langle e^2(iK)\rangle \triangleq \frac{1}{K}\sum_{q=0}^{K-1} e^2(iK+q), \quad (3)$$

and $\hat{\nabla}\varepsilon$ is also an estimate of the gradient of the mean squared error

$$\nabla\varepsilon(iK) = \nabla E\{e^2(iK)\} = -2E\{e(iK)\boldsymbol{X}(iK)\}. \quad (4)$$

In these expressions, $e(i)$ is the estimation error

$$e(i) = d(i) - \hat{d}(i). \quad (5)$$

For the special case $K = 1$, the SGD algorithm is the well known LMS algorithm.[2]

[2] It is well known that $K = 1$ yields the most rapidly converging algorithm, if convergence rate is measured in units of data time-samples. However, if convergence rate is measured in units of algorithm iterations (which are K times less frequent than data time-samples) then, speed of convergence *increases* monotonically with K. Thus, in applications (such as satellite antenna array adaptation using microwave power dividers) for which adjustment of the weight vector can be a time-consuming costly task, $K > 1$ is preferable. Also, as shown in this paper, steady-state misadjustment is approximately proportional to K^{-1}. Consequently, in applications for which misadjustment can be undesirably large (e.g., because of data corruption, as explained in Section 2.D), $K > 1$ is preferable (cf. [7]). Furthermore, regardless of desirability of $K > 1$ in practice, an analysis that includes K as a parameter reveals important relationships between stochastic-data algorithms and non-stochastic-data algorithms, since the latter can be obtained from the former by letting $K \to \infty$, as exploited in this paper. Moreover, as explained at the end of Section 2.B, the effects of the *degree of randomness* of the stochastic gradient are explicitly parameterized by K. As a final remark, it should be mentioned that algorithms which average the gradient over K instants, but adjust the weight vector every instant, do not perform as well as the SGD algorithm (with either $K = 1$ or $K > 1$) studied herein.

B.1. Applications

Before continuing, we briefly review five types of applications in order to give physical meaning to the mathematical quantities (1)–(5) to be dealt with in the analysis. The five tasks of system identification, noise filtering, noise cancelling, system equalization, and signal prediction are depicted in Figs. 1–5. With regard to Fig. 1, the task is to identify the unknown system. This is accomplished by adaptively adjusting W to make the estimate $\hat{d}$ similar to the *desired* quantity d, in which case W should be similar to the unknown system (n is measurement noise). With regard to Fig. 2, the task is to reduce the noise n. This is accomplished by adaptively adjusting W to make the estimate $\hat{d}$ similar to the desired signal d, thereby filtering away the noise, n. With regard to Fig. 3, the task is to reduce the noise d. This is accomplished by adaptively adjusting W to make the estimate $\hat{d}$ similar to d, and then subtracting this noise estimate from the noise corrupted signal $\tilde{d}$, in an attempt to cancel the noise d (s is the noiseless signal). With regard to Fig. 4, the task is to remove the distortion of the signal d caused by the unknown system. This is accomplished by adaptively adjusting W to make the estimate $\hat{d}$ similar to d, thereby making W similar to the inverse of the unknown system (n is noise due, for example, to decision errors in decision-directed adaptation). With regard to Fig. 5, the task is to predict the value of the process x, i_0 units of time into the future. This is accomplished by adaptively adjusting W in the lower signal path to make the delayed (by i_0) estimate $\hat{d}(i)$ similar to the delayed desired signal $d(i)$, and then slaving the non-adaptive W in the upper signal path to the adaptive W. In all five of these applications, the only quantities that are physically available for use in adaptation are x, $\hat{d}$, $\tilde{d}$, and therefore $\tilde{e} = \tilde{d} - \hat{d}$. All other quantities, viz., s and n, are unavailable.

It should be clarified that, as shown in Figs. 1, 3, 4, the error signal $\tilde{e}$ used to adjust W is the difference between $\tilde{d}$ and $\hat{d}$, not the difference between d and $\hat{d}$; that is $\tilde{d}$, which is physically available, is used in place of the desired d, which

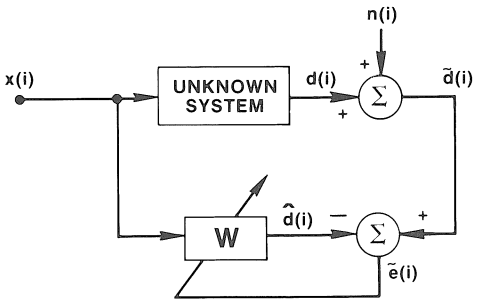

Fig. 1. System identification.

is not physically available. The analysis in the body of this paper is carried out in terms of $e = d - \hat{d}$. Then in Section 2.D, the modifications of results that are required when $\tilde{e} = \tilde{d} - \hat{d}$ is used in place of e, for adjustment of W, are explained. In all five problems described here, the vector $\boldsymbol{X}(i)$ in (1) has nth element $x_n(i) = x(i-n)$. But this is not the case for adaptive sensor array problems, and is not assumed to be the case in this paper, except in Section 3C.2.

B.2. Simplifying assumptions

As discussed in Section 1.A, it is assumed that $\{\boldsymbol{X}(i), d(i)\}$ is an independent identically distributed sequence of zero-mean pairs, and this shall be referred to as the *primary independence assumption*[3]. It is also assumed that the vectors $\boldsymbol{X}(i)$ have finite fourth moments, and the scalars $d(i)$ have finite second moments. The autocovariance matrix for $\boldsymbol{X}(i)$ and the crosscovariance vector for $\boldsymbol{X}(i)$

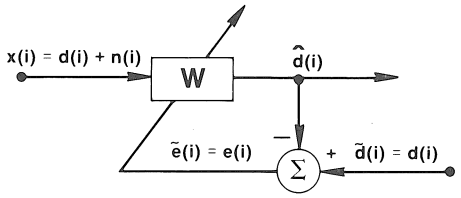

Fig. 2. Noise filtering.

[3] For the important case of $x_n(i) = x(i-n)$, the primary independence assumption is literally impossible, unless the algorithm is slowed down by a sufficiently large factor, say L, so that the sequence $\boldsymbol{X}(i)$ is replaced with the sequence $\boldsymbol{X}(iL)$.

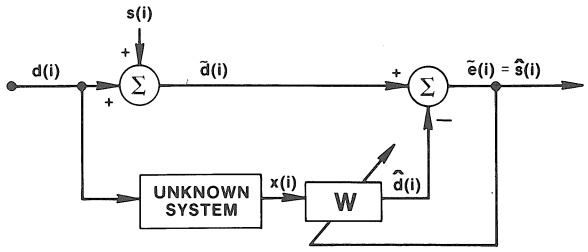

Fig. 3. Noise cancelling.

and $d(i)$ are denoted by

$$\boldsymbol{R} \triangleq E\{\boldsymbol{X}(i)\boldsymbol{X}^{\mathrm{T}}(i)\},$$
$$\boldsymbol{P} \triangleq E\{\boldsymbol{X}(i)d(i)\}, \tag{6}$$

and the variance of $d(i)$ is denoted by

$$\sigma_d^2 = E\{d^2(i)\}. \tag{7}$$

It is well known and easily shown that the weight vector that minimizes the mean-squared-error (MSE), which is denoted by

$$\varepsilon(i) \triangleq E\{e^2(i)\}, \tag{8}$$

is given by

$$\boldsymbol{W}_0 = \boldsymbol{R}^{-1}\boldsymbol{P}, \tag{9}$$

and the corresponding minimum value of $\varepsilon(i)$ is

$$\varepsilon_0 \triangleq E\{e_0^2(i)\}$$
$$= \sigma_d^2 - \boldsymbol{P}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{P}, \tag{10}$$

where

$$e_0(i) \triangleq d(i) - \boldsymbol{W}_0^{\mathrm{T}}\boldsymbol{X}(i). \tag{11}$$

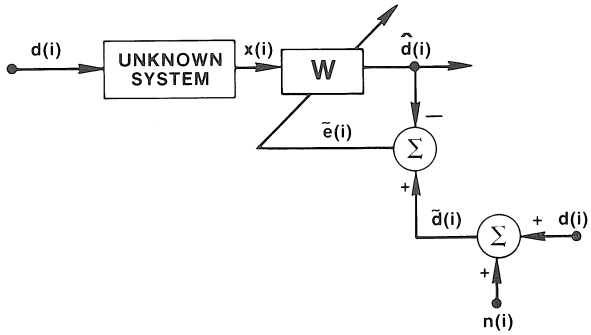

Fig. 4. System equalization.

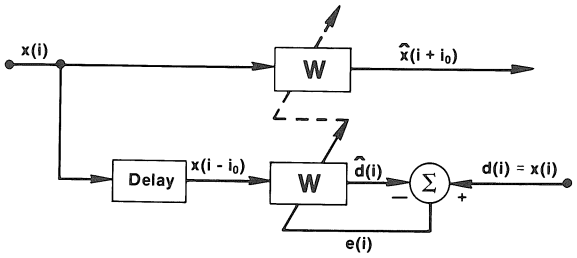

Fig. 5. Signal prediction.

The behavior of the SGD algorithm shall be characterized in terms of the *learning curve*, which is defined to be the evolution of the *excess* MSE, which is defined by

$$\varepsilon_*(i) = \varepsilon(i) - \varepsilon_0. \tag{12}$$

In order to obtain the simplest explicit formulas for learning curves the *secondary independence assumption* that the *minimum attainable error* $e_0(i)$ is independent of the training vector $\boldsymbol{X}(i)$ is employed. This assumption is satisfied, for example, if the desired sequence, $d(i)$, is of the form

$$d(i) = \tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{X}(i) + z(i) \tag{13}$$

for some non-random vector $\tilde{\boldsymbol{W}}$ and for some random sequence $z(i)$ that is independent of $\boldsymbol{X}(i)$ (since, in this case, $z(i) = e_0(i)$). Model (13) is valid for the problem of adaptive identification of an unknown finite-impulse-response system ($\tilde{\boldsymbol{W}}$) of order $\leq N$, with measurement noise $z(i)$ that is independent of the probe signal $\boldsymbol{X}(i)$ (e.g., Fig. 1, with $z(i) = n(i)$). Model (13) is also valid for the problem of noiseless adaptive equalization of an all-pole channel with $\leq N$ poles (e.g., Fig. 4), and for the problem of noise cancellation involving an all-pole reference channel with $\leq N$ poles (e.g., Fig. 3). Finally, the secondary independence assumption is valid, regardless of model (13), if $\boldsymbol{X}(i)$ and $d(i)$ are jointly Gaussian sequences (as in Section 3.C), because the uncorrelatedness of $e_0(i)$ and $\boldsymbol{X}(i)$ (which follows from the orthogonality property of $\hat{d}_0(i) = \boldsymbol{W}_0^{\mathrm{T}}\boldsymbol{X}(i)$) then renders $e_0(i)$ and $\boldsymbol{X}(i)$ independent. In applications for which neither model (13) nor the Gaussian model are valid, the secondary independence assumption must be considered an *approximation* akin to the primary independence assumption.

2. Effects of randomness, particular distribution, and corruption of data on learning characteristics

2.A. Learning characteristics

It is well known, and easily shown (using the primary independence assumption only), that the excess MSE is given by

$$\varepsilon_*(i) = E\{\boldsymbol{V}^{\mathrm{T}}(i)\boldsymbol{R}\boldsymbol{V}(i)\} = \mathrm{tr}\{\boldsymbol{R}_V(i)\boldsymbol{R}\}, \tag{14}$$

where $\boldsymbol{V}(i)$ is the *weight-vector error* (or excess weight-vector)

$$\boldsymbol{V}(i) \triangleq \boldsymbol{W}(i) - \boldsymbol{W}_0, \tag{15}$$

and $\boldsymbol{R}_V(i)$ is its correlation matrix. Manipulation of the SGD algorithm (2) yields

$$\begin{aligned} &\boldsymbol{V}([i+1]K) = \boldsymbol{A}(iK)\boldsymbol{V}(iK) + \boldsymbol{B}(iK) \\ &\boldsymbol{V}(iK+q) = \boldsymbol{V}(iK), \quad q = 1, 2, \ldots, K-1, \end{aligned} \tag{16}$$

where

$$\begin{aligned} \boldsymbol{A}(iK) &\triangleq I - \frac{\mu}{K}\sum_{q=0}^{K-1}\boldsymbol{X}(iK+q)\boldsymbol{X}^{\mathrm{T}}(iK+q), \\ \boldsymbol{B}(iK) &\triangleq \frac{\mu}{K}\sum_{q=0}^{K-1} e_0(iK+q)\boldsymbol{X}(iK+q). \end{aligned} \tag{17}$$

Substitution of (16)–(17) into (14) (evaluated at $i \to (i+1)K$), and use of both the primary and secondary independence assumptions yields the exact formula

$$\begin{aligned} \varepsilon_*([i+1]K) &= E\{\boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{G}\boldsymbol{R}\boldsymbol{V}(iK)\} + h \\ &= \mathrm{tr}\{\boldsymbol{R}_V(i)\boldsymbol{G}\boldsymbol{R}\} + h, \\ \varepsilon_*(iK+q) &= \varepsilon_*(iK), \quad q = 1, 2, \ldots, K-1, \end{aligned} \tag{18}$$

where

$$h \triangleq E\{\boldsymbol{B}^{\mathrm{T}}(iK)\boldsymbol{B}(iK)\} = \varepsilon_0(\mu\lambda_{\mathrm{rms}})^2 N/K, \tag{19}$$

and

$$G \triangleq E\{\boldsymbol{A}^{\mathrm{T}}(iK)\boldsymbol{A}(iK)\}\boldsymbol{R}^{-1}$$
$$= (\boldsymbol{I}-\mu\boldsymbol{R})^2+\mu^2\boldsymbol{S}/K, \qquad (20)$$
$$\boldsymbol{S} \triangleq E\{\boldsymbol{X}(i)\boldsymbol{X}^{\mathrm{T}}(i)\boldsymbol{R}\boldsymbol{X}(i)\boldsymbol{X}^{\mathrm{T}}(i)\}\boldsymbol{R}^{-1}-\boldsymbol{R}^2.$$

In (19), λ_{rms} is the *root-mean-square* value of the N eigenvalues of $\boldsymbol{R}$:

$$\lambda_{\mathrm{rms}} \triangleq \left[\frac{1}{N}\sum_{n=1}^{N}\lambda_n^2\right]^{1/2} = \left[\frac{1}{N}\mathrm{tr}\{\boldsymbol{R}^2\}\right]^{1/2}. \qquad (21)$$

Formulas (14) and (18)–(20) are the point of focus for discussion of the effects of randomness of the gradient, the effects of the particular distribution of the data, and the effects of corruption to the desired training signal, on the characteristics of the learning curve. The characteristics of primary concern are:

(i) *Stability*: The largest value (supremum) of the fixed step-size parameter μ that yields a stable algorithm is denoted by μ_*:

$$\lim_{i\to\infty}\varepsilon_*(i) \quad \text{exists if and only if} \quad 0<\mu<\mu_*. \qquad (22)$$

(ii) *Final Misadjustment*: The fractional amount by which the steady state MSE exceeds the minimum attainable MSE is called the final misadjustment and is denoted by M:

$$M \triangleq \lim_{i\to\infty}\varepsilon_*(i)/\varepsilon_0. \qquad (23)$$

(iii) *Initial Rate of Convergence*: The initial rate of convergence is defined to be the inverse of an *effective initial time constant* (τ), which is obtained by fitting $\varepsilon_*(iK)$ (at $i=0$ and $i=1$) to an exponential (cf. (46)):

$$[\varepsilon_*(0)-\varepsilon_*(\infty)]\,\mathrm{e}^{-K/\tau}=[\varepsilon_*(K)-\varepsilon_*(\infty)];$$

thus,

$$\frac{1}{\tau} \triangleq \frac{1}{K}\ln\{[\varepsilon_*(0)-\varepsilon_*(\infty)]/[\varepsilon_*(K)-\varepsilon_*(\infty)]\}. \qquad (24)$$

(iv) *Optimum Step-Size Sequence*: The sequence of step-size parameters that minimizes the instantaneous MSE at every adaptation instant is denoted by $\mu_0(iK)$:

$$\varepsilon_*(iK)|_{\mu=\mu_0(iK)} \leqslant \varepsilon_*(iK)|_{\mu=\mu(iK)} \qquad (25)$$

for all sequences $\mu(iK)$.

2.B. Isolation of effects of data randomness

Formula (18)–(20) can be used to isolate the effects on the learning curve of randomness of the data and therefore the stochastic gradient. Since explicit formulas for exact learning characteristics exist when randomness vanishes, this isolation aids analysis of the effects of randomness on characteristics of the learning curve derived in this paper.

With reference to (2), it can be seen that in the limit (as the gradient-averaging time-interval becomes large) $K\to\infty$, the stochastic squared error (3) and its gradient (2) approach non-random quantities (i.e., they approach their mean values because of ergodicity guaranteed by the primary independence assumption), and the SGD algorithm becomes simply a non-stochastic *gradient-descent* (GD) algorithm, for which the adapting weight vector is non-random. As a result, for sufficiently large values of K, the expectation operator in (14) and (18) can be deleted, and (14), (18)–(20) become

$$\varepsilon_*(iK) = \boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{R}\boldsymbol{V}(iK), \qquad (14)'$$
$$\varepsilon_*([i+1]K) = \boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{G}\boldsymbol{R}\boldsymbol{V}(iK)+h, \qquad (18)'$$

where

$$h=0, \qquad \boldsymbol{G}=(\boldsymbol{I}-\mu\boldsymbol{R})^2. \qquad (19)', (20)'$$

It is well known and easily verified that these equations reduce (with the aid of (16)) to

$$\varepsilon_*(iK) = \boldsymbol{V}^{\mathrm{T}}(0)\boldsymbol{G}^i\boldsymbol{R}\boldsymbol{V}(0). \qquad (26)$$

Equation (26) can be decomposed into a sum of the N natural modes of the learning curve by transforming the vector $\boldsymbol{V}$ with the orthogonal matrix $\boldsymbol{Q}$ composed of the eigenvectors of $\boldsymbol{R}$,

$$\tilde{\boldsymbol{V}} \triangleq \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{V}. \qquad (27)$$

The result is

$$\varepsilon_*(iK) = \sum_{n=1}^{N} \varepsilon_{*n}(0)(1-\mu\lambda_n)^{2i}$$

$$= \sum_{n=1}^{N} \varepsilon_{*n}(0)\, e^{-iK/\tau_n}, \tag{28}$$

where the nth time-constant is defined by

$$\tau_n \triangleq K\{\ln(1/\gamma_n)\}^{-1},$$
$$\gamma_n \triangleq (1-\mu\lambda_n)^2, \tag{29}$$

and

$$\varepsilon_{*n}(0) = \tilde{v}_n^2 \lambda_n, \tag{30}$$

which is the component of the initial excess MSE due to the nth mode.

(i) *Stability*: It follows from (28) that the largest step-size for which the GD algorithm is stable is

$$\mu_* = 2/\lambda_{\max}, \tag{31}$$

where $\lambda_{\max}$ is the largest of the N eigenvalues $\{\lambda_n\}_1^N$.

(ii) *Final Misadjustment*: It follows from (28) that the final misadjustment for the GD algorithm is zero,

$$M = 0. \tag{32}$$

(iii) *Initial Rate of Convergence*: It follows from (28) that the initial rate of convergence for the GD algorithm is

$$\frac{1}{\tau} = \frac{1}{K} \ln\left\{ \frac{\sum_{n=1}^{N} \varepsilon_{*n}(0)}{\sum_{n=1}^{N} \varepsilon_{*n}(0)(1-\mu\lambda_n)^2} \right\}. \tag{33}$$

We see that the effective initial time constant, τ, depends on the initial distribution of excess MSE among the N modes. We therefore define the *nominal time constant* to be that which results from a uniform distribution of initial weight-vector error among the N uncoupled modes (i.e., $\tilde{v}_n^2 = v^2$);

Then (30) and (33) yield

$$\frac{1}{\tau_{\text{nom}}} = \frac{1}{K} \ln\left\{ \frac{\sum_{n=1}^{N} \lambda_n}{\sum_{n=1}^{N} \lambda_n(1-\mu\lambda_n)^2} \right\}$$

$$= \frac{1}{K} \ln\left\{ \frac{\text{tr}\{\boldsymbol{R}\}}{\text{tr}\{\boldsymbol{R}(\boldsymbol{I}-\mu\boldsymbol{R})^2\}} \right\}$$

$$= \frac{1}{K} \ln\left\{ \frac{\text{tr}\{\boldsymbol{R}\}}{\text{tr}\{\boldsymbol{G}\boldsymbol{R}\}} \right\}. \tag{34}$$

In the special case of slow adaptation ($\mu \ll 1/\lambda_n$), (34) is closely approximated by

$$\frac{1}{\tau_{\text{nom}}} \simeq \frac{2\mu}{K} \frac{\text{tr}\{\boldsymbol{R}^2\}}{\text{tr}\{\boldsymbol{R}\}}. \tag{35}$$

An intuitively pleasing interpretation of formula (35) results from re-expression in terms of the parameters λ_{rms} (21) and

$$\lambda_{\text{ave}} \triangleq \frac{1}{N} \text{tr}\{\boldsymbol{R}\} = \frac{1}{N} \sum_{n=1}^{N} \lambda_n, \tag{36}$$

for which

$$\lambda_{\text{rms}}^2 = \sigma_\lambda^2 + \lambda_{\text{ave}}^2, \tag{37}$$

where σ_λ^2 is the *variance of the distribution* $\{\lambda_n\}_1^N$ about its *mean* λ_{ave}. Substitution of (36) and (37) into (35) yields

$$\frac{1}{\tau_{\text{nom}}} = \left(\frac{2\mu\lambda_{\text{ave}}}{K}\right)(1+\rho), \tag{38}$$

$$\rho \triangleq \sigma_\lambda^2/\lambda_{\text{ave}}^2, \tag{39}$$

for which ρ is a *normalized measure of the spread of the distribution* $\{\lambda_n\}_1^N$. Hence, the larger the spread is, the faster the algorithm is (initially) *relative to the conservative estimate of speed based on* λ_{ave} *alone.* The estimate,

$$\frac{1}{\tau_{\text{conserv}}} \triangleq \frac{2\mu\lambda_{\text{ave}}}{K}, \tag{40}$$

is conservative because it ignores the fact that the slower modes (smaller λ_n) have smaller weights as revealed by (28) and (30). Nevertheless, the slower modes eventually dominate, so that the larger the spread is, the slower the algorithm is in terms of reaching steady state.

(iv) *Optimum Step-Size Sequence*: The optimum step-size sequence for the GD algorithm can be obtained by equating to zero the derivative of $\varepsilon_*([i+1]K)$ in (18)′, with respect to μ, which yields

$$\mu_0(iK) = \frac{\boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{R}^2\boldsymbol{V}(iK)}{\boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{R}^3\boldsymbol{V}(iK)},$$

Use of the bound

$$\begin{aligned}\lambda_{\min}\boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{R}^2\boldsymbol{V}(iK) &\leq \boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{R}^3\boldsymbol{V}(iK)\\ &\leq \lambda_{\max}\boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{R}^2\boldsymbol{V}(iK)\end{aligned}$$

in the demoninator yields

$$\mu_{\min} \leq \mu_0(iK) \leq \mu_{\max},$$

$$\mu_{\min} \triangleq 1/\lambda_{\max}, \qquad \mu_{\max} \triangleq 1/\lambda_{\min}. \tag{41}$$

Another approach to maximization of speed of convergence is to minimize the nominal time constant τ_{nom} (34), with respect to μ. This results in

$$\mu^0_{\mathrm{nom}} = \mathrm{tr}\{R^2\}/\mathrm{tr}\{R^3\}, \tag{42}$$

and

$$\tau^0_{\mathrm{nom}} = K\left\{\ln\left(\frac{1}{1-\alpha}\right)\right\}^{-1},$$

$$\alpha \triangleq \mu^0_{\mathrm{nom}}\,\mathrm{tr}\{\boldsymbol{R}^2\}/\mathrm{tr}\{\boldsymbol{R}\} = \mu^0_{\mathrm{nom}}\lambda_{\mathrm{ave}}(1+\rho). \tag{43}$$

It should be noted that μ^0_{nom} is within the bounds (41).

In conclusion, the essence of the difference between (18) for the SGD algorithm and (18)′ for the GD algorithm is that $E\{\cdot\}$ vanishes in (18)′. In addition, h vanishes in (18)′, and the form of $\boldsymbol{G}$ is different in (18)′ than it is in (18) because the term $\mu^2\boldsymbol{S}/K$ in (20) vanishes in (20)′. However, the fact that the form of $\boldsymbol{G}$ is different has no bearing on the existence of an explicit solution of the form

$$\varepsilon_*(iK) = \boldsymbol{V}^{\mathrm{T}}(0)\boldsymbol{G}^i\boldsymbol{R}\boldsymbol{V}(0),$$

although the form of $\boldsymbol{G}$ does determine the rates of convergence (time-constants) of the N exponential modes which comprise $\varepsilon_*(iK)$. Similarly, the presence of a non-zero h in (18)′ would simply make the linear recursion non-homogeneous, in which case $M \neq 0$. In contrast to these minor differences, the presence of the expectation operator $E\{\cdot\}$ in (18) smears the simple evolutionary law, (18)′, and in effect introduces a coupling between modes that cannot be removed by a transformation of coordinates (as in (27)). Nevertheless, in the special case for which the elements of $\boldsymbol{X}(i)$ are jointly Gaussian, a partial decoupling can be accomplished, and is directly responsible for the explicit formulas, in terms of N coupled recursions, for exact characteristics of the learning curve derived in Section 3.C. Unfortunately, no decoupling of modes can be accomplished in the general case of non-Gaussian data. And it is for this reason that the exact solution involves N^2 (actually, only $(N^2+N)/2$, due to symmetry) coupled linear recursions, as revealed in Section 3.B.

As a final remark, it is mentioned that the effects of the *degree of randomness* of the stochastic gradient in the SGD algorithm are explicitly characterized in Sections 2 and 3, through the explicit dependence of learning characteristics on the gradient-averaging time-interval, K.

2.C. Isolation of effects of particular data distribution

The only parameter in formula (18) that depends on the distribution of $\boldsymbol{X}(i)$, through more than just the covariance matrix $\boldsymbol{R}$, is the matrix $\boldsymbol{S}$, which depends on the fourth joint moments of $\boldsymbol{X}(i)$. This matrix can have a significant effect on all of the characteristics, (i)–(iv), of the learning curve. To illustrate the nature of the effects of the distribution of the data, without undue complication, the case of independent identically distributed (i.i.d.) elements of the training vector is considered.

If the elements of the vector $\boldsymbol{X}(i)$ are independent and have identical even distributions, with

variance and kurtosis[4] denoted by σ_x^2 and ν_x, then h and the matrices $\boldsymbol{R}$ and $\boldsymbol{G}$ in (14) and in (18)–(20) reduce to

$$\boldsymbol{R} = \sigma_x^2 \boldsymbol{I}, \qquad \boldsymbol{G} = \gamma \boldsymbol{I}, \tag{44}$$

$$\gamma = 1 - 2\mu\sigma_x^2 + \mu^2\sigma_x^4\left(1 + \frac{N + \nu_x - 2}{K}\right),$$

$$h = \varepsilon_0(\mu\sigma_x^2)^2 N/K.$$

Substitution of (14) and (44) into (18) yields

$$\varepsilon_*([i+1]K) = \gamma\varepsilon_*(iK) + h, \tag{45}$$

which can be solved to obtain

$$\varepsilon_*(iK) = [\varepsilon_*(0) - \varepsilon_*(\infty)]\, e^{-iK/\tau} + \varepsilon_*(\infty), \tag{46}$$

where

$$\varepsilon_*(\infty) = h/(1-\gamma), \tag{47}$$

$$\tau = K\{\ln(1/\gamma)\}^{-1}. \tag{48}$$

(i) *Stability*: It follows from (45) that the SGD algorithm (2) is stable if and only if

$$|\gamma| < 1. \tag{49}$$

Substitution of (44) into (49) yields the following maximum value (supremum) of the step-size parameter for which the SGD algorithm is stable

$$\mu_* = 2\left[\sigma_x^2\left(1 + \frac{N + \nu_x - 2}{K}\right)\right]^{-1}. \tag{50}$$

Hence, distributions with large kurtosis, ν_x, require small step-size for stability. Similarly, the larger the parameter ratio N/K is, the smaller the step-size must be. This should be contrasted with (31) for the GD algorithm, for which stability is independent of the parameters ν_x, N, K. We note however, that (31) is the limit of (50) as $K \to \infty$.

(ii) *Final Misadjustment*: It follows from (47) that the misadjustment for the SGD algorithm is

$$M = \frac{\eta}{1 - \eta(1 + [K + \nu_x - 2]/N)}, \tag{51}$$

$$\eta \triangleq \mu\sigma_x^2 N/2K. \tag{52}$$

[4] The kurtosis, ν_x, for a zero-mean variable, x, is defined by $\nu_x = E\{x^4\}/(E\{x^2\})^2$. For example, $\nu_x = \infty$ for the Cauchy distribution, $\nu_x = 1$—its minimum value—for a binary symmetric distribution, and $\nu_x = 3$ for a Gaussian distribution.

Hence distributions with large kurtosis require smaller step-size for a given misadjustment. Similarly, the larger the parameter ratio N/K is, the larger the misadjustment is. This should be contrasted with the fact that $M = 0$ (32) for the GD algorithm. We note, however, that (32) is the limit of (51) as $K \to \infty$.

(iii), (iv) *Initial Rate of Convergence, and Optimum Step Size*: The initial rate of convergence $(1/\tau)$ is given by (48). This rate is as high as possible when γ is as small as possible. It follows from (44) that the minimum value of γ is

$$\gamma_0 = 1 - \mu_0\sigma_x^2, \tag{53}$$

and is achieved with the step-size $\mu = \mu_0$

$$\mu_0 = \left[\sigma_x^2\left(1 + \frac{N + \nu_x - 2}{K}\right)\right]^{-1} = \mu_*/2. \tag{54}$$

Formula (54) should be contrasted with (41) and (42), for the GD algorithm, which reduce (using (44)) to

$$\mu_0 = 1/\sigma_x^2. \tag{55}$$

Hence, for the SGD algorithm, the initial optimum step-size is smaller by the factor K/N (for $K/N \ll 1$).

Substitution of (54) into (53) yields

$$\gamma_0 = \frac{(N + \nu_x - 2)/K}{1 + (N + \nu_x - 2)/K}, \tag{56}$$

Hence, maximum initial speed of convergence is low for distributions with large kurtosis. Similarly, the larger the parameter ratio N/K is, the lower the maximum initial rate of convergence is. We note that for large values of $(N + \nu_x - 2)/K$, (56) is closely approximated by

$$\gamma_0 \simeq 1 - K/(N + \nu_x - 2), \tag{57}$$

in which case

$$\tau_0 \simeq N + \nu_x - 2. \tag{58}$$

2.D. Isolation of effects of data corruption

As briefly discussed in Section 1.B, there are many applications for which the desired signal d

is unavailable for adaptation but for which a related (corrupted) signal $\tilde{d}$ is available instead (cf. Figs. 1, 3, 4). As a result, the error signal e (5) is unavailable for use in the SGD algorithm (2), and the related (corrupted) error signal

$$\tilde{e}(i) \triangleq \tilde{d}(i) - \hat{d}(i) \tag{59}$$

must be used in its place. If an interference (corruption) signal δ is defined by

$$\delta(i) \triangleq \tilde{d}(i) - d(i) \tag{60}$$

(e.g., $\delta = n$ in Figs. 1, 4, and $\delta = s$ in Fig. 3), then the error signals $\tilde{e}$ and e are related by

$$\tilde{e}(i) = e(i) + \delta(i). \tag{61}$$

If $\delta(i)$ and $\boldsymbol{X}(i)$ are orthogonal (as they are, for example, in the applications depicted in Figs. 1 and 3), then the GD algorithm is unaffected by the interference δ, because $E\{\tilde{e}(i)\boldsymbol{X}(i)\} = E\{e(i)\boldsymbol{X}(i)\}$ in (4). On the other hand, the SGD algorithm (2) is indeed affected by δ, regardless of possible orthogonality or even independence between $\delta(i)$ and $\boldsymbol{X}(i)$.

The effects of δ can be determined simply by replacing e with $\tilde{e} = e + \delta$ everywhere that e occurs in the analysis in Sections 2.C and 3. However, since e affects only the quantities ε_* and ε_0 in these results, then we need make only the replacements

$$\begin{aligned} &\varepsilon_0 \to \tilde{\varepsilon}_0, \\ &\varepsilon_*(i) \to \tilde{\varepsilon}_*(i) \triangleq \tilde{\varepsilon}(i) - \tilde{\varepsilon}_0, \end{aligned} \tag{62}$$

where

$$\begin{aligned} &\tilde{\varepsilon}(i) \triangleq E\{\tilde{e}^2(i)\}, \\ &\tilde{\varepsilon}_0(i) \triangleq E\{\tilde{e}_0(i)\}, \end{aligned} \tag{63}$$

and $\tilde{e}_0$ is defined parallel to (11) (with d replaced by $\tilde{d}$), provided that it is assumed (parallel to the primary independence assumption) that $\delta(i)$ is a zero-mean sequence of independent elements. A significantly simplifying assumption that is justifiable for many applications is that $\delta(i)$ is orthogonal to both $\boldsymbol{X}(i)$ and $d(i)$. In this case,

$$\tilde{\varepsilon}_0 = \varepsilon_0 + \sigma_\delta^2 \tag{64}$$

and

$$\tilde{\varepsilon}_*(i) = \varepsilon_*(i) + 2E\{e(i)\delta(i)\}, \tag{65}$$

where σ_δ^2 is the variance of the interference $\delta(i)$. Furthermore, since $\delta(i)$ is orthogonal to $e(i)$, then (65) reduces to

$$\tilde{\varepsilon}_*(i) = \varepsilon_*(i), \tag{66}$$

from which it follows that

$$\tilde{\varepsilon}(i) = \varepsilon(i) + \sigma_\delta^2. \tag{67}$$

Therefore, the dynamics of $\varepsilon(i)$ are the same as those of $\tilde{\varepsilon}(i)$. Hence, the only modifications to be made to the analysis in Sections 2.C and 3 follow from the appropriate modification to the constant driving term h in the learning curve equation (18); i.e., h gets multiplied by the *degradation factor*, D,

$$D \triangleq \tilde{\varepsilon}_0 / \varepsilon_0 = 1 + \sigma_\delta^2 / \varepsilon_0. \tag{68}$$

The consequences of this are:

(i) *Stability*: unaffected

(ii) *Final Misadjustment*: M is replaced with $\tilde{M}$, which is given by formulas in Sections 2.C and 3. But the misadjustment factor of real interest is still M, not $\tilde{M}$, and

$$M = \tilde{M}D, \tag{69}$$

where $\tilde{M}$ is given by the formulas for M in Sections 2.C and 3.

(iii) *Initial Rate of Convergence*: unaffected

(iv) *Optimum Step-Size Sequence*: The instantaneous misadjustment $M(i) = \varepsilon_*(i)/\varepsilon_0$ is replaced with

$$\tilde{M}(i) = \tilde{\varepsilon}_*(i)/\tilde{\varepsilon}_0 = M(i)/D \tag{70}$$

in the formulas for $\mu_0(i)$ in Section 3, which leaves the initial optimum step-size unaffected, but multiplies the final step-size arithmetic sequence by the attentuation factor $1/D$.

In the special case for which $\varepsilon_0 = 0$, we cannot divide by ε_0 as in (68)–(70); however, the appropriate counterpart of the result (69) is

$$\lim_{i \to \infty} \varepsilon(i) = \tilde{M}\sigma_\delta^2, \tag{71}$$

where $\tilde{M}$ is given by formulas for M in Sections 2.C and 3.

3. General formulas for learning characteristics

In this section, formulas giving exact evaluations, approximate evaluations, and bounds on learning curves and characteristics are derived.

3.A. Bounds on characteristics

The simplicity of the preceding analysis (Section 2.C) of the first-order, linear, time-invariant recursion (45) for the learning curve for the special case of i.i.d. elements of the training vector, $\boldsymbol{X}(i)$, motivates the approach of seeking bounds on the learning curve (18), for the general case, that are of the form (45).

It is well known that a quadratic form such as that in (18) can be bounded by another quadratic form as follows

$$\begin{aligned}\gamma_{\min} E\{\boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{R}\boldsymbol{V}(iK)\} \\ \leq E\{\boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{G}\boldsymbol{R}\boldsymbol{V}(iK)\} \\ \leq \gamma_{\max} E\{\boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{R}\boldsymbol{V}(iK)\},\end{aligned} \tag{72}$$

where γ_m, $m = \min$ and max, are the minimum and maximum eigenvalues of the matrix $\boldsymbol{H}$,

$$\boldsymbol{H} \triangleq \tfrac{1}{2}(\boldsymbol{G}+\boldsymbol{G}^{\mathrm{T}}) = (\boldsymbol{I}-\mu\boldsymbol{R})^2 + \mu^2(\boldsymbol{S}+\boldsymbol{S}^{\mathrm{T}})/2K. \tag{73}$$

Both $\gamma_{\min}$ and $\gamma_{\max}$ are non-negative since $\boldsymbol{G}$ is non-negative definite. Substitution of (72) and (14) into (18) yields

$$\varepsilon_*([i+1]K) \underset{m=\min}{\overset{m=\max}{\leqslant}} \gamma_m \varepsilon_*(iK) + h. \tag{74}$$

It follows from (74) that the learning curve is bounded above and below by

$$\varepsilon_{\min}(iK) \leqslant \varepsilon_*(iK) \leqslant \varepsilon_{\max}(iK), \tag{75}$$

where

$$\begin{aligned}\varepsilon_m([i+1]K) &= \gamma_m \varepsilon_m(iK) + h, \\ \varepsilon_m(0) &= \varepsilon_*(0),\end{aligned} \tag{76}$$

for $m = \min, \max$.

It follows from (76) that

$$\varepsilon_m(iK) = [\varepsilon_*(0) - \varepsilon_m(\infty)]\mathrm{e}^{-iK/\tau_m} + \varepsilon_m(\infty), \tag{77}$$

where

$$\varepsilon_m(\infty) = h/(1-\gamma_m), \tag{78}$$

$$\tau_m \triangleq K\{\ln(1/\gamma_m)\}^{-1}. \tag{79}$$

(i) *Stability*: Since the exact learning curve must be stable ($\lim \varepsilon_*(i)$, $i \to \infty$, exists), if its upper bound is stable, then a sufficient (but not necessary) condition for stability is

$$|\gamma_{\max}| < 1. \tag{80}$$

Similarly, a necessary (but not sufficient) condition for stability is

$$|\gamma_{\min}| < 1. \tag{81}$$

(ii) *Final Misadjustment*: It follows from (72)–(78) and (19) that the exact final misadjustment (23) for the SGD algorithm is bounded by

$$M_{\min} \leqslant M \leqslant M_{\max}, \tag{82}$$

$$M_m \triangleq \frac{(\mu\lambda_{\mathrm{rms}})^2 N/K}{1-\gamma_m}. \tag{83}$$

(iii) *Initial Rate of Convergence*: It follows from (75)–(79) that the exact effective initial time constant (24) is bounded by

$$\tau_{\min} \leqslant \tau \leqslant \tau_{\max}, \tag{84}$$

$$\tau_m \triangleq K\{\ln(1/\gamma_m)\}^{-1}. \tag{85}$$

(iv) *Optimum Step-Size Sequence*: Since the optimum sequence of step-sizes for a bound on the learning curve is not, in general, a bound (or even a useful approximation) to the optimum sequence of step-sizes for the exact learning curve, we cannot use the bounds (77) to investigate step-size optimization. However, quadratic-form bounds analogous to (72) can be used to obtain bounds on the exact optimum step-size sequence as follows. In order to minimize $\varepsilon_*([i+1]K)$ at each adjustment instant, iK, the derivative of (18) (which is quadratic in μ), w.r.t. μ, is equated to

zero to obtain the condition

$$\mu_0(iK) = \frac{E\{\boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{R}^2\boldsymbol{V}(iK)\}}{E\{\boldsymbol{V}^{\mathrm{T}}(iK)[\boldsymbol{R}^2+\boldsymbol{S}/K]\boldsymbol{R}\boldsymbol{V}(iK)\}+\varepsilon_0\lambda_{\mathrm{rms}}^2 N/K}, \tag{86}$$

which is implicit since $\boldsymbol{V}(iK)$ depends on $\mu_0([i-1]K)$. Substitution of (86) into (18) yields

$$\varepsilon_*([i+1]K) = \varepsilon_*(iK) - \mu_0(iK)E\{\boldsymbol{V}^{\mathrm{T}}(iK)\boldsymbol{R}^2\boldsymbol{V}(iK)\}. \tag{87}$$

Now, use of bounds analogous to (72) in both the numerator and denominator of (86) yields

$$\mu_{\min}^0(iK) \leq \mu_0(iK) \leq \mu_{\max}^0(iK), \tag{88}$$

$$\mu_{\max}^0(iK) \triangleq \frac{\lambda_{\max}M(iK)}{\delta_{\min}M(iK)+\lambda_{\mathrm{rms}}^2 N/K} \tag{89}$$

(and similarly for $\mu_{\min}^0(iK)$), where $M(iK)$ is the *instantaneous misadjustment*, defined by

$$M(iK) \triangleq \varepsilon_*(iK)/\varepsilon_0, \tag{90}$$

and $\delta_{\min}$ is the minimum eigenvalue of the matrix $\boldsymbol{D}$

$$\boldsymbol{D} \triangleq \boldsymbol{R}^2+(\boldsymbol{S}+\boldsymbol{S}^{\mathrm{T}})/2K. \tag{91}$$

Furthermore, by bounding the quadratic form in (87), the following bounds on the exact step-size-optimized learning curve, $\varepsilon_*^0(iK)$, are obtained

$$[1-\lambda_{\max}\mu_0(iK)]\varepsilon_*^0(iK) \leq \varepsilon_*^0([i+1]K) \leq [1-\lambda_{\min}\mu_0(iK)]\varepsilon_*^0(iK). \tag{92}$$

Consider now the two extremes of *initial* and *final* behavior.

Initially, $M(iK) \gg 1$, and (89) reduces (since δ_m typically has order of magnitude $\lambda_{\mathrm{rms}}^2 N/K$, which—for example—is easily verified for Gaussian $\boldsymbol{X}(i)$) to

$$\mu_{\max}^0(iK) \simeq \lambda_{\max}/\delta_{\min}, \tag{93}$$

and similarly for $\mu_{\min}^0(iK)$. (For the special case considered in Section 2.C, (93) reduces to (54).) Use of bound (93) in (92) yields

$$\varepsilon_*^0(0)(1-\lambda_{\max}^2/\delta_{\min})^i \leq \varepsilon_*^0(iK) \leq \varepsilon_*^0(0)(1-\lambda_{\min}^2/\delta_{\max})^i, \tag{94}$$

from which it follows that the minimum initial time-constant is bounded by (84)–(85), with $\gamma_{\min}$ and $\gamma_{\max}$ replaced with $\gamma_{\min}^0$ and $\gamma_{\max}^0$,

$$\gamma_{\min}^0 \triangleq 1-\lambda_{\max}^2/\delta_{\min}. \tag{95}$$

Finally, $M(iK) \ll 1$, and the counterpart of (89) reduces to

$$\mu_{\min}(iK) \simeq (\lambda_{\min}/\lambda_{\mathrm{rms}}^2)(N/K)\left(\frac{1}{\varepsilon_0}\right)\varepsilon_*^0(iK). \tag{96}$$

Use of (96) in (92) yields

$$\varepsilon_*^0([i+1]K) \leq [1-c_1\varepsilon_*^0(iK)]\varepsilon_*^0(iK), \tag{97}$$

where c_1 is a constant. But since

$$\varepsilon_*^0([i+1]K) \leq \varepsilon_*^0(iK), \tag{98}$$

(97) implies

$$\varepsilon_*^0([i+1]K) \leq \{1-c_1\varepsilon_*^0([i+1]K)\}\varepsilon_*^0(iK), \tag{99}$$

which is equivalent to

$$\varepsilon_*^0([i+1]K) \leq \frac{\varepsilon_*^0(iK)}{1+c_1\varepsilon_*^0(iK)}. \tag{100}$$

The solution to (100), with the inequality replaced with equality, yields the upper bound

$$\varepsilon_*^0(iK) \leq (c_2+c_1 iK)^{-1} \leq (c_1 iK)^{-1}. \tag{101}$$

Use of (101) in the counterpart of (96) yields an upper bound on $\mu_{\max}^0(iK)$ and therefore on $\mu_0(iK)$, viz.,

$$\mu_0(iK) \leq c_3/iK. \tag{102}$$

It is concluded that both the exact optimum step-size and the exact excess MSE decrease at least arithmetically in the final stages of convergence. This should be contrasted with the optimum step-size sequence for the GD algorithm. As revealed by (41), the speed-optimized GD algorithm does not exhibit an arithmetically decreasing step-size (because its final misadjustment is zero for a fixed step-size).

For the special case of independent elements of the vector $\boldsymbol{X}(i)$, which is addressed in Section 2.C, all upper and lower bounds (preceding (98) in this

Section (3.A)) coincide to give exact formulas. However, the stronger the dependence among the elements of the vector $\boldsymbol{X}(i)$ is, the more disparate the upper and lower bounds in this section are. This is a result of the well-known fact that as the correlation among a set of variables increases, the ratio of the maximum-to-minimum eigenvalues of the corresponding correlation matrix increases. As a result, the quantitative utility of these bounds, when used directly, diminishes (although their qualitative interpretations can remain useful) as dependence increases. For this reason, exact formulas for learning curves are developed in the next section.

3.B. Exact solution for learning curve

It follows from (16), using both the primary and secondary independence assumptions, that the excess weight correlation matrix,

$$\boldsymbol{R}_V(iK) \triangleq E\{\boldsymbol{V}(iK)\boldsymbol{V}^{\mathrm{T}}(iK)\}, \tag{103}$$

satisfies the linear recursion

$$\boldsymbol{R}_V([i+1]K) = E\{\boldsymbol{A}(iK)\boldsymbol{R}_V(iK)\boldsymbol{A}(iK)\} + \frac{\mu^2\varepsilon_0}{K}\boldsymbol{R}, \tag{104}$$

$$\boldsymbol{R}_V(iK+q) = \boldsymbol{R}_V(iK), \quad q = 1,2,3,\ldots,k-1.$$

Expressed more explicitly in terms of the (nm)th element of $\boldsymbol{R}_V$, (104) becomes

$$[\boldsymbol{R}_V([i+1]K)]_{nm} = \sum_{j,k=1}^{N} M_{nmkj}[\boldsymbol{R}_V(iK)]_{kj} + \frac{\mu^2\varepsilon_0}{K}[\boldsymbol{R}]_{nm}, \tag{105}$$

where

$$M_{nmkj} \triangleq E\{[\boldsymbol{A}(iK)]_{nk}[\boldsymbol{A}(iK)]_{jm}\}. \tag{106}$$

By concatenating the column vectors of each of the matrices $\boldsymbol{R}$, $\boldsymbol{R}_V(iK)$, they each can be re-interpreted as (column) vectors in an N^2-dimensional space, and the $N^2 \times N^2$ array with elements M_{nmjk} can be interpreted as a linear operator, (matrix), $\boldsymbol{M}$, on this space. Then (105) can be re-expressed as the N^2-dimensional vector, first order, linear, time-invariant recursion

$$\boldsymbol{R}_V([i+1]K) = \boldsymbol{M}\boldsymbol{R}_V(iK) + \frac{\mu^2\varepsilon_0}{K}\boldsymbol{R}. \tag{107}$$

The solution to this recursion is

$$\boldsymbol{R}_V(iK) = \boldsymbol{M}^i\boldsymbol{R}_V(0) + \frac{\mu^2\varepsilon_0}{K}[\boldsymbol{I}-\boldsymbol{M}]^{-1}[\boldsymbol{I}-\boldsymbol{M}^i]\boldsymbol{R}, \tag{108}$$

which has the steady state value

$$\boldsymbol{R}_V(\infty) = \frac{\mu^2\varepsilon_0}{K}[\boldsymbol{I}-\boldsymbol{M}]^{-1}\boldsymbol{R}. \tag{109}$$

This formula (108), together with (14) yields the desired formula for the exact learning curve $\varepsilon_*(i)$. With the aid of a computer, this learning curve formula can be used to graph learning curves, and thereby graphically study learning characteristics.

Unfortunately, this formula requires products with and inversion of an $N^2 \times N^2$ matrix (e.g., 10^8 matrix elements for a 100 weight filter or a 100 element antenna). A considerable simplification of this solution, that involves a matrix of dimension only $N \times N$, occurs in the special case for which $\boldsymbol{X}(i)$ is Gaussian, as revealed in the next section.

3.C. Explicit formulas for the case of gaussian data[5]

If the elements of the data vector $\boldsymbol{X}(i)$ are jointly Gaussian (with zero mean), then the fourth-joint-

[5] The unpublished work of Kenneth D. Senne [19] is closely related to some of the results in Section 3.C.1. Specifically, explicit equations and formulas (125), (138), (145), (152) can be derived (with some algebraic manipulation) from Senne's equations. Some of these derivations were carried out by Horowitz and Senne (independently of the author's work, which was not based on [19]), and are reported in [20] (which was published 6 months after the author's first disclosure). All results in Section 3.C.1 and in [20] (for real weight vectors) follow from (111) as shown herein, and (111) follows from (104) simply by exploiting the fourth-moment decomposition (110) for Gaussian variables. From this point of view, the results in [19] follow from the author's more general linear recursion (104) by invoking the Gaussian assumption.

moment decomposition,

$$E\{x_n(i)x_m(i)x_p(i)x_q(i)\} = E\{x_n(i)x_m(i)\}E\{x_p(i)x_q(i)\} + E\{x_n(i)x_p(i)\}E\{x_m(i)x_q(i)\} + E\{x_n(i)x_q(i)\}E\{x_m(i)x_p(i)\}, \quad (110)$$

can be used (together with (17), and the primary independence assumption) to simplify the first term on the right side of (104),

$$E\{\boldsymbol{A}(iK)\boldsymbol{R}_V(iK)\boldsymbol{A}(iK)\} = (\boldsymbol{I}-\mu\boldsymbol{R})\boldsymbol{R}_V(iK)(\boldsymbol{I}-\mu\boldsymbol{R}) + \frac{\mu^2}{K}[\boldsymbol{R}\boldsymbol{R}_V(iK)\boldsymbol{R} + \mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_V(iK)\}\boldsymbol{R}]. \quad (111)$$

Substitution of (111) into (104) yields

$$\boldsymbol{R}_V([i+1]K) = \boldsymbol{R}_V(iK) - \mu\boldsymbol{R}_V(iK)\boldsymbol{R} - \mu\boldsymbol{R}\boldsymbol{R}_V(iK) + \mu^2\frac{K+1}{K}\boldsymbol{R}\boldsymbol{R}_V(iK)\boldsymbol{R} + \frac{\mu^2}{K}[\varepsilon_0 + \mathrm{tr}\{\boldsymbol{R}\boldsymbol{R}_V(iK)\}]\boldsymbol{R}. \quad (112)$$

C.1. Learning curve and characteristics.

Equation (112) is still an N^2-vector linear, time-invariant recursion in the N^2 elements of the matrix $\boldsymbol{R}_V(iK)$. However, it yields an N-vector recursion for $\varepsilon_*(iK)$ through use of (14), which is repeated here

$$\varepsilon_*(iK) = \mathrm{tr}\{\boldsymbol{R}_V(iK)\boldsymbol{R}\}. \quad (113)$$

(Also, (112) yields an N-vector recursion for $E\{\|\boldsymbol{V}(iK)\|^2\}$, through use of $E\{\|\boldsymbol{V}(iK)\|^2\} = \mathrm{tr}\{\boldsymbol{R}_V(iK)\}$.) Specifically, let $\boldsymbol{Q}$ be the orthogonal matrix composed of the eigenvectors of $\boldsymbol{R}$. Then

$$\boldsymbol{R} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}}, \quad (114)$$

where $\boldsymbol{\Lambda} = \mathrm{diag}\{\boldsymbol{\lambda}\}$ is the diagonal matrix whose N-vector of diagonal elements,

$$\boldsymbol{\lambda} = \{\lambda_n\}_1^N, \quad (115)$$

is composed of the eigenvalues of $\boldsymbol{R}$. Define the non-diagonal (in general) matrix $\boldsymbol{\Gamma}(iK)$ by

$$\boldsymbol{\Gamma}(iK) \triangleq \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{R}_V(iK)\boldsymbol{Q}, \quad (116)$$

and denote the N-vector of diagonal elements of $\boldsymbol{\Gamma}(iK)$ by

$$\boldsymbol{\gamma}(iK) = \{\gamma_n(iK)\}_1^N. \quad (117)$$

Substitution of (114) and (116) into (112) yields the following N-vector recursion for the diagonal elements of $\boldsymbol{\Gamma}(iK)$ (the off-diagonal elements, which are not of concern here, are specified by an N^2-vector recursion)

$$\boldsymbol{\gamma}([i+1]K) = \boldsymbol{F}\boldsymbol{\gamma}(iK) + \frac{\mu^2\varepsilon_0}{K}\boldsymbol{\lambda}, \quad (118)$$

for which

$$\boldsymbol{F} \triangleq \boldsymbol{I} - 2\mu\boldsymbol{\Lambda} + \left(\frac{K+1}{K}\right)\mu^2\boldsymbol{\Lambda}^2 + \frac{\mu^2}{K}\boldsymbol{\lambda}\boldsymbol{\lambda}^{\mathrm{T}}. \quad (119)$$

Let $\boldsymbol{U}$ be the orthogonal matrix composed of the eigenvectors of $\boldsymbol{F}$. Then

$$\boldsymbol{F} = \boldsymbol{U}\,\mathrm{diag}\{\boldsymbol{f}\}\boldsymbol{U}^{\mathrm{T}}, \quad (120)$$

where $\mathrm{diag}\{\boldsymbol{f}\}$ is the diagonal matrix whose N-vector of diagonal elements, $\boldsymbol{f} = \{f_n\}_1^N$, is composed of the eigenvalues of $\boldsymbol{F}$. Substitution of (120) into (118) yields

$$\tilde{\gamma}_n([i+1]K) = f_n\tilde{\gamma}_n(iK) + \frac{\mu^2\varepsilon_0}{K}\tilde{\lambda}_n, \quad (121)$$

where

$$\tilde{\boldsymbol{\gamma}} \triangleq \boldsymbol{U}^{\mathrm{T}}\boldsymbol{\gamma}, \qquad \tilde{\boldsymbol{\lambda}} \triangleq \boldsymbol{U}^{\mathrm{T}}\boldsymbol{\lambda}. \quad (122)$$

The solution to (121) is

$$\tilde{\gamma}_n(iK) = \tilde{\gamma}_n(0)f_n^i + \frac{\mu^2\varepsilon_0}{K}\tilde{\lambda}_n\left[\frac{1-f_n^i}{1-f_n}\right]. \quad (123)$$

By expressing the geometric progression in terms of an exponential

$$f_n^i = \mathrm{e}^{-iK/\tau_n}, \quad (124)$$

the *time constants* of evolution of $\gamma(iK)$,

$$\tau_n \triangleq K[\ln(1/f_n)]^{-1}, \tag{125}$$

are identified. Substitution of (114), (116), (123), (124) into (112) yields the desired formula for the learning curve:

$$\varepsilon_*(iK) = \sum_{n=1}^{N} [\varepsilon_n(0) - \varepsilon_n(\infty)]\, e^{-iK/\tau_n} + \varepsilon_*(\infty), \tag{126}$$

for which

$$\varepsilon_n(0) = \tilde{\lambda}_n \tilde{\gamma}_n(0), \tag{127}$$

$$\varepsilon_n(\infty) = \frac{\varepsilon_0 \mu^2 \tilde{\lambda}_n^2 / K}{1 - f_n}, \tag{128}$$

and

$$\varepsilon_*(\infty) = \sum_{n=1}^{N} \varepsilon_n(\infty), \tag{129}$$

$$\varepsilon_*(0) = \sum_{n=1}^{N} \varepsilon_n(0). \tag{130}$$

To obtain an explicit formula for the steady state excess MSE, $\varepsilon_*(\infty)$, it is easiest to start with (14),

$$\begin{aligned} \varepsilon_*(\infty) &= \mathrm{tr}\{\boldsymbol{R}_V(\infty)\boldsymbol{R}\} \\ &= \mathrm{tr}\{\boldsymbol{\Gamma}(\infty)\boldsymbol{\Lambda}\} \\ &= \boldsymbol{\gamma}^{\mathrm{T}}(\infty)\boldsymbol{\lambda} = \sum_{n=1}^{N} \gamma_n(\infty)\lambda_n \qquad (131) \\ &= \tilde{\boldsymbol{\gamma}}^{\mathrm{T}}(\infty)\tilde{\boldsymbol{\lambda}} = \sum_{n=1}^{N} \tilde{\gamma}_n(\infty)\tilde{\lambda}_n. \qquad (132) \end{aligned}$$

Substitution of $i+1 = i = \infty$ in (121) yields

$$\tilde{\gamma}_n(\infty) = \frac{\varepsilon_0 \mu^2 \tilde{\lambda}_n / K}{1 - f_n}, \tag{133}$$

which, together with (132), verifies (129). Now, let us use (131), and proceed to evaluate $\gamma_n(\infty)$. Substituting $i+1 = i = \infty$ in (118) yields

$$\boldsymbol{\gamma}(\infty) = \frac{\mu^2 \varepsilon_0}{K} (\boldsymbol{I} - \boldsymbol{F})^{-1} \boldsymbol{\lambda}. \tag{134}$$

Since $\boldsymbol{I} - \boldsymbol{F}$ is the sum of a diagonal matrix and a rank-one matrix, it can easily be inverted using *Woodbury's Identity* (e.g. [14, p. 655]) to obtain

$$\gamma_n(\infty) = \frac{\mu \varepsilon_0}{2K} \left(\frac{1}{1-\eta}\right)\left(\frac{1}{1-\beta_n}\right), \tag{135}$$

for which

$$\begin{aligned} \eta &= \frac{\mu}{2K} \sum_{n=1}^{N} \frac{\lambda_n}{1 - ([K+1]/2K)\mu\lambda_n} \\ &= \frac{\mu}{2K} \mathrm{tr}\left\{ \boldsymbol{R}\left[\boldsymbol{I} - \frac{K+1}{2K}\mu\boldsymbol{R} \right]^{-1} \right\}, \qquad (136) \end{aligned}$$

and

$$\beta_n = \frac{K+1}{2K} \mu\lambda_n. \tag{137}$$

Substitution of (135)–(137) into (131) yields

$$\varepsilon_*(\infty) = \varepsilon_0 \frac{\eta}{1-\eta}, \tag{138}$$

which is the desired result.

To obtain a *linear-system* interpretation of the time constants, $\{\tau_n\}_1^N$, it is easiest to again start with (14)

$$\begin{aligned} \varepsilon_*(iK) &= \mathrm{tr}\{\boldsymbol{R}_V(iK)\boldsymbol{R}\} \\ &= \mathrm{tr}\{\boldsymbol{\Gamma}(iK)\boldsymbol{\Lambda}\} \\ &= \boldsymbol{\gamma}^{\mathrm{T}}(iK)\boldsymbol{\lambda} = \sum_{n=1}^{N} \lambda_n \gamma_n(iK). \qquad (139) \end{aligned}$$

Now, let us express equations (118) and (139) in standard *unity-feedback state-variable* form [14]

$$\begin{aligned} &\boldsymbol{Z}(i+1) = \boldsymbol{A}\boldsymbol{Z}(i) + \boldsymbol{B}[u(i) + y(i)], \\ &y(i) = \boldsymbol{C}\boldsymbol{Z}(i), \qquad (140) \end{aligned}$$

where the system state is

$$\boldsymbol{Z}(i) = \boldsymbol{\gamma}(iK), \tag{141}$$

the system input and output are

$$u(i) = \varepsilon_0, \qquad y(i) = \varepsilon_*(iK), \tag{142}$$

and the system matrices are

$$\begin{aligned} &\boldsymbol{A} = F - BC = \boldsymbol{I} - 2\mu\boldsymbol{\Lambda} + \frac{K+1}{K}\mu^2\boldsymbol{\Lambda}^2, \\ &\boldsymbol{B} = \frac{\mu^2}{K}\boldsymbol{\lambda}, \\ &\boldsymbol{C} = \boldsymbol{\lambda}^{\mathrm{T}}. \qquad (143) \end{aligned}$$

Since $\boldsymbol{A}$ is diagonal, the *open-loop states* are uncoupled. A signal-flow diagram of this system is shown in Fig. 6. It follows from the conventional *transfer-function* formula for a unity-feedback state-variable model that this system has transfer function [14]

$$H(z)=\frac{\boldsymbol{C}(\boldsymbol{A}-z\boldsymbol{I})^{-1}\boldsymbol{B}}{1-\boldsymbol{C}(\boldsymbol{A}-z\boldsymbol{I})^{-1}\boldsymbol{B}}, \tag{144}$$

from which the system poles (and corresponding time constants, $\{\tau_n\}_1^N$) can be obtained as the roots of the denominator polynomial, which are simply the eigenvalues of the matrix $\boldsymbol{A}+\boldsymbol{BC}=\boldsymbol{F}$, in (119).

(i) *Stability*: The SGD algorithm is stable if and only if $\varepsilon_*(\infty)<\infty$. It follows from (138) that instability sets in, as the stepsize (μ) is increased, when $\eta=1$. The corresponding value of μ is the solution, μ_*, to $\eta=1$, which can be expressed (using (136)) as

$$\mu_*=\frac{2K}{\mathrm{tr}\{\boldsymbol{R}[\boldsymbol{I}-([K+1]/2K)\mu_*\boldsymbol{R}]^{-1}\}}. \tag{145}$$

An explicit approximation for this stability bound is

$$\mu_*\simeq\frac{2K}{\mathrm{tr}\{\boldsymbol{R}\}}. \tag{146}$$

To determine the conditions under which (146) is a close approximation, (146) is substituted into (145) to obtain

$$\frac{2K}{\mathrm{tr}\{\boldsymbol{R}\}}\simeq\frac{2K}{\mathrm{tr}\{\boldsymbol{R}[\boldsymbol{I}-([K+1]/\mathrm{tr}\{\boldsymbol{R}\})\boldsymbol{R}]^{-1}\}},$$

which is, apparently[6], a close approximation if

$$(K+1)\ll\frac{(\mathrm{tr}\{\boldsymbol{R}\})^2}{\mathrm{tr}\{\boldsymbol{R}^2\}}. \tag{147}$$

Since the right side is upper-bounded by N, then $N/K\gg 1$ is sufficient for (146) to be accurate.

(ii) *Final misadjustment*: It follows from (23) and (138), that the final misadjustment is

$$M=\frac{\eta}{1-\eta}. \tag{148}$$

A commonly used (small-μ)-approximation (cf. [17, Chapt. 4, and references therein]) for M is

$$M\simeq\frac{\mu}{2K}\mathrm{tr}\{\boldsymbol{R}\}. \tag{149}$$

It follows, parallel to the argument in the preceding paragraph[6], that this is a good approximation if both

$$\mu\ll\frac{2K}{\mathrm{tr}\{\boldsymbol{R}\}}\left(\frac{N}{K+1}\right)$$

[6] This approximation can be verified by expanding the inverse matrix into a power series in the matrix $([K+1]/\mathrm{tr}\{\boldsymbol{R}\})\boldsymbol{R}$.

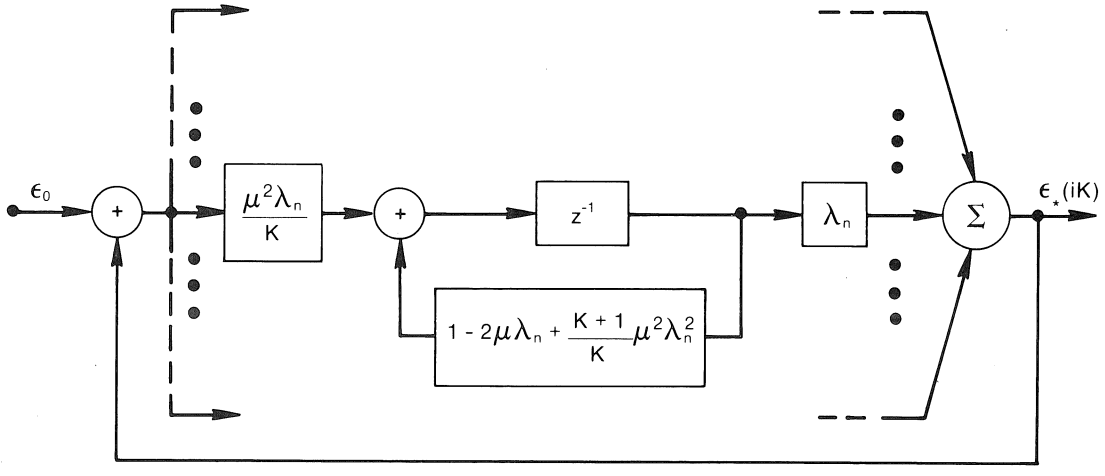

Fig. 6. Signal-flow diagram of linear system model of learning curve dynamics.

and

$$\mu \ll \frac{2K}{\mathrm{tr}\{\boldsymbol{R}\}}. \tag{150}$$

C.2. Steady state behavior of weight vector

In order to obtain an explicit formula for the steady-state autocorrelation matrix, $\boldsymbol{R}_V(\infty)$, for the weight-error vector, $\boldsymbol{V}(i)$ defined in (15), it would appear that the N^2-vector recursion (112) must be solved. However, the non-diagonal matrix $\boldsymbol{\Gamma}(iK)$ is asymptotically ($i \to \infty$) diagonal so that again we need the solution for only the N-vector recursion (118). Substitution of (135)–(137) into

$$\boldsymbol{R}_V(\infty) = \boldsymbol{Q}\boldsymbol{\Gamma}(\infty)\boldsymbol{Q}^{\mathrm{T}} \tag{151}$$

yields

$$\boldsymbol{R}_V(\infty) = \frac{\mu\varepsilon_0}{2K}\left(\frac{1}{1-\eta}\right)\left[\boldsymbol{I} - \frac{K+1}{2K}\mu\boldsymbol{R}\right]^{-1}. \tag{152}$$

This solution is easily verified by substitution into (112) with $i = i+1 = \infty$.

A commonly used (small$-\mu$)-approximation (cf. [15], [16], and [17] and references therein) for $\boldsymbol{R}_V(\infty)$ is

$$\boldsymbol{R}_V(\infty) \simeq \frac{\mu\varepsilon_0}{2K}\boldsymbol{I}. \tag{153}$$

It follows from (152) that (153) is an accurate approximation if μ is small enough to satisfy (150).

In applications of the SGD algorithm to spectral analysis problems, the spectral characteristics of the weight vector are of interest (e.g. [15]). It is known (e.g. [6]) that in steady state,

$$\boldsymbol{W}(i) = \boldsymbol{W}_0 + \boldsymbol{V}(i), \tag{154}$$

where $\boldsymbol{W}_0$ is, of course, non-random and $\boldsymbol{V}(i)$ has zero steady-state mean. Therefore the steady-state autocovariance of $\boldsymbol{V}(i)$ is identical to its steady-state autocorrelation (152). It is now assumed that the elements of $\boldsymbol{X}(i)$ are $x_n(i) = x(i-n)$, as discussed in Section 1.B.1. It follows from (9), and the theory of discrete-time Wiener filtering (and/or the theory of asymptotics of toeplitz matrices [21]–[23]) that the discrete Fourier transform (DFT) of the sequence of mean weights, $w_0(1), w_0(2), w_0(3), \ldots, w_0(N)$ is approximated by

$$\mathrm{DFT}\{\boldsymbol{W}_0\} \simeq S_{dx}(f)/S_x(f), \tag{155}$$

where S_{dx} is the cross spectral density of $d(i)$ and $x(i)$ (from which $\boldsymbol{X}(i)$ is derived as $\{x(i), x(i-1), \ldots, x(i-N+1)\}$), and $S_x(f)$ is the power spectral density of $x(i)$. The approximation is close for N much greater than the correlation time of $x(i)$, and becomes exact as $N \to \infty$. Furthermore, in steady state, the elements $\{v_j(i)\}_1^N$ of $\boldsymbol{V}(i)$ are correlated stationary random processes with time index i. Moreover, with i fixed the N-vector $\boldsymbol{V}(i) = \{v_1(i), v_2(i), \ldots, v_N(i)\} = \{v_j(i)\}_1^N$ can be viewed as a segment of a random process with index j. The autocovariance of this zero-mean process is

$$R_{v(i)}(j,k) = E\{v_j(i)v_k(i)\} = [\boldsymbol{R}_V(i)]_{jk}, \tag{156}$$

and is the (jk)th element of $\boldsymbol{R}_V(i)$. Since $\boldsymbol{R}$ is a Toeplitz matrix, it follows from (152) that $\boldsymbol{R}_V(i)$ is, in steady state, the inverse of a Toeplitz matrix. Thus for sufficiently large N (N much larger than the correlation time of the process $x(i)$), $\boldsymbol{R}_V(i)$ is approximately Toeplitz

$$R_{v(i)}(j,k) \simeq R_{v(i)}(j-k). \tag{157}$$

This approximation becomes exact as $N \to \infty$. Hence the process $v_j(i)$ (with i fixed) is approximately wide-sense stationary, and its power spectral density is (for $N \to \infty$) the DFT

$$S_{v(i)}(f) = \sum_{l=-\infty}^{\infty} R_{v(i)}(l)\, \mathrm{e}^{-\sqrt{-1}\,2\pi f l}. \tag{158}$$

It follows from (136), (152), (157), (158), and the theory of asymptotics of toeplitz matrices [21]–[23] that

$$S_{v(i)}(f) = \frac{\mu\varepsilon_0}{2K}\left(\frac{1}{1-\eta}\right) \times \left[\frac{1}{1-([K+1]/2K)\mu S_r(f)}\right], \tag{159}$$

where

$$\eta = \frac{\mu}{2K} \int_{-1/2}^{1/2} \frac{S_x(f)}{1-(K+1/2K)\mu S_x(f)}\, df. \tag{160}$$

Formulas (155) and (159)–(160) comprise the desired spectral characterization of the steady-state weight vector for large N. It follows from (159) that for sufficiently small values of μ, $S_{v(i)}(f)$ is accurately approximated by a constant, independent of frequency, f, in which case the weight process is white.

Finally, the physical significance of $S_{v(i)}(f)$ is that

$$E\left\{\frac{1}{N}|\mathrm{DFT}\{\boldsymbol{V}(i)\}|^2\right\} \simeq S_{v(i)}(f); \tag{161}$$

i.e., the left side of (161) is a smoothed version of the right side (with smoothing-window width $1/N$), provided that N is sufficiently large to validate (157). This is a standard result from the spectral theory of stationary processes.

3.D. Approximate misadjustment formula for non-Gaussian data

It follows from (14) and (18) that in steady state

$$\varepsilon_*(\infty) = \mathrm{tr}\{\boldsymbol{R}_V(\infty)\boldsymbol{R}\} = \mathrm{tr}\{\boldsymbol{R}_V(\infty)\boldsymbol{G}\boldsymbol{R}\} + h, \tag{162}$$

which can be manipulated into the form

$$M = \frac{\varepsilon_*(\infty)}{\varepsilon_0} = \frac{\left(\frac{\mu^2}{K}\right)\mathrm{tr}\{\boldsymbol{R}^2\}}{1-\frac{\mathrm{tr}\{\boldsymbol{R}_V(\infty)\boldsymbol{G}\boldsymbol{R}\}}{\mathrm{tr}\{\boldsymbol{R}_V(\infty)\boldsymbol{R}\}}}. \tag{163}$$

In obtaining this formula, no simplifying assumptions other than the primary and secondary independence assumptions (to obtain (162)) were used. From this characterization for M, the upper and lower bounds (83), in terms of the extreme eigenvalues of $\boldsymbol{G}+\boldsymbol{G}^{\mathrm{T}}$, can be obtained directly. As an alternative, motivated by the approximation (153) (cf. [15], [16]), which yields

$$\frac{\mathrm{tr}\{\boldsymbol{R}_V(\infty)\boldsymbol{G}\boldsymbol{R}\}}{\mathrm{tr}\{\boldsymbol{R}_V(\infty)\boldsymbol{R}\}} \simeq \frac{\mathrm{tr}\{\boldsymbol{G}\boldsymbol{R}\}}{\mathrm{tr}\{\boldsymbol{R}\}}, \tag{164}$$

M can be approximated, from (163) and (164) by

$$M \simeq \frac{(\mu^2/K)\mathrm{tr}\{\boldsymbol{R}^2\}}{1-\mathrm{tr}\{\boldsymbol{G}\boldsymbol{R}\}/\mathrm{tr}\{\boldsymbol{R}\}}. \tag{165}$$

Use of (20) yields

$$\mathrm{tr}\{\boldsymbol{G}\boldsymbol{R}\} = \mathrm{tr}\{(\boldsymbol{I}-\mu\boldsymbol{R})^2\boldsymbol{R}\} + \frac{\mu^2}{K}\mathrm{tr}\{\boldsymbol{S}\boldsymbol{R}\}, \tag{166}$$

which upon substitution into (165) yields

$$M \simeq \frac{M_0}{1-M_0\left[\frac{\mathrm{tr}\{\boldsymbol{S}\boldsymbol{R}\}+K\,\mathrm{tr}\{\boldsymbol{R}^3\}}{\mathrm{tr}\{\boldsymbol{R}\}\mathrm{tr}\{\boldsymbol{R}^2\}}\right]}, \tag{167}$$

$$M_0 \triangleq \frac{\mu}{2K}\mathrm{tr}\{\boldsymbol{R}\}. \tag{168}$$

Substitution of (20) for $\boldsymbol{S}$ into (167) yields

$$M \simeq \frac{M_0}{1-M_0\left[\frac{\mathrm{tr}\{[\boldsymbol{K}_X+(K-1)\boldsymbol{I}]\boldsymbol{R}^3\}}{\mathrm{tr}\{\boldsymbol{R}\}\mathrm{tr}\{\boldsymbol{R}^2\}}\right]}, \tag{169}$$

where $\boldsymbol{K}_X$ is the *kurtosis matrix*

$$\boldsymbol{K}_X \triangleq E\{[\boldsymbol{X}(i)\boldsymbol{X}^{\mathrm{T}}(i)]^2\}[E\{\boldsymbol{X}(i)\boldsymbol{X}^{\mathrm{T}}(i)\}]^{-2}. \tag{170}$$

As an example, if the elements of $\boldsymbol{X}(i)$ are i.i.d. (non-Gaussian in general), then (164) is exact, and consequently (169) reduces to the exact formula (51)–(52). On the other hand, if $\boldsymbol{X}(i)$ is Gaussian, then the approximation (164) can be shown to agree closely with the exact formula (148) only if μ is sufficiently small to render (149) an accurate approximation.

4. Critique

4.A. Independence assumption

It is important to emphasize that the useful characterization (14) of excess MSE is, in general, invalid if $\boldsymbol{X}(i)$ and $d(i)$ are not each independent sequences. In [6, Secs. VII, IX] and [3, p. 547], (14) is used without this independence assumption. As a result, attempts in [6, Sec. X] and [3, p. 551] to verify the applicability of theoretical learning

curve characteristics (derived on the basis of the independence assumption) to situations where the independence assumption is violated, proceed by substitution of simulated samples of $\boldsymbol{V}(iK)$ into (14) to evaluate $\varepsilon_*(iK)$. Hence, the independence assumption is inadvertently invoked, thereby invalidating the verification procedure.

4.B. *Effects of data randomness*

The important matrix $\boldsymbol{S}$ in (20), which vanishes only for the GD algorithm (or $K \to \infty$) and for binary symmetric data $x_n(i) = \pm 1$ (so that $\boldsymbol{K}_x = \boldsymbol{I}$ in (170)), is omitted from most previous investigations of SGD algorithms, either by consideration of only sufficiently small values of μ (so that terms in μ^2 are negligible compared with terms in μ) [5], [6], [17, chapter 4], or through use of apparently inaccurate—in general—approximations; e.g., $E\{x^4\} \simeq (E\{x^2\})^2$ [3, *p.* 555], or $E\{[\boldsymbol{X}(i)\boldsymbol{X}^{\mathrm{T}}(i)]^2\} \simeq [E\{\boldsymbol{X}(i)\boldsymbol{X}^{\mathrm{T}}(i)\}]^2$ [7, p. 838], or $\boldsymbol{X}^{\mathrm{T}}(i)\boldsymbol{X}(i) \simeq NE\{x^2\}$ [9, p. 307].

4.C. *Stability*

It is well known that the LMS algorithm requires a relatively small step-size for adequate stability. It would therefore appear that analyses of the LMS algorithm that make simplifying small-step-size approximations are adequate (cf. [17, chapter 4] and references therein). However, the more general SGD algorithm ($K > 1$) can be stable for a relatively large step size. In fact, in applications for which $K \gg 1$ is desirable (see footnote 2), a good tradeoff between rate of convergence and misadjustment requires a relatively large step size.

The conditions (80) and (81) yield stability bounds on μ that contradict the bound in [9, p. 310], where it is concluded (by using a combination of a bound and an approximation) that $\mu_* < 2/N\lambda_{\max}$ (for $K = 1$). The analysis in Section 3.C for Gaussian $\boldsymbol{X}(i)$ and large N yields stability bound (146), which corroborates [10, (46)]; however the Gaussian assumption is not made in [10], but rather terms in μ^2 are discarded by assuming μ_* is small, and by making an apparently inaccurate—in general—approximation of a fourth moment [10, Appendix].

4.D. *Misadjustment*

The exact formulas (148) and (163) contradict [7, p. 838] where it is concluded that M is proportional to $1/K$. The dependence on K in the denominators in (148) and (163) vanishes in [7] due to use of the apparently inaccurate—in general—approximation $E\{[\boldsymbol{X}(i)\boldsymbol{X}^{\mathrm{T}}(i)]^2\} \simeq [E\{\boldsymbol{X}(i)\boldsymbol{X}^{\mathrm{T}}(i)\}]^2$. In addition, (148) and (163) are not in agreement with [4, (22)–(24)], which is obtained through use of an assumed bound on the norm of $\boldsymbol{W}(i)$ [4, (24c)]. Also, neither the bounds (82)–(83) nor the exact formulas (148) and (163) are in complete agreement with [10, (47)], where it is concluded (by using an apparently innacurate—in general—approximation [10, Appendix]) that misadjustment is given by $M \simeq (\mu/2)\,\mathrm{tr}\{\boldsymbol{R}\}$ (for $K = 1$). Nevertheless, [10, (47)] is valid for Gaussian $\boldsymbol{X}(i)$ and large N (cf. (149)), although the Gaussian assumption is not made in [10]. The exact explicit formula (148) reduces to the exact implicit (in terms of eigenvalues or diagonalized $\boldsymbol{R}$) formula in [18], for the special case $K = 1$.

4.E. *Optimum step-size*

Previous investigations seek characterizations of the optimum step-size and corresponding maximum rate of convergence by optimizing μ for an upper bound on $\varepsilon_*(iK)$, [4, pp. 127, 129], or a combination upperbound and approximation [9, pp. 307, 310]. In contrast, Section 3.A provides bounds on the optimum quantities, rather than optimum quantities for bounds. The bound (93) does not, in general, agree with [4, (31)], where it is concluded that $\mu_0 = \lambda_{\min}/\lambda_{\max}^2$ (for $K = 1$); nor does it agree with [9, p. 310], where it is concluded that $\mu_0 = 1/N\lambda_{\max}$. On the other hand, the bound (101) corroborates [4, (35)] (which is obtained by approximation of inequality [4, (30)] with inequality [4, (34)] in which some, but not all, terms of the same order of magnitude are discarded).

4.F. Time to convergence

As revealed in Section 2.B for the GD algorithm, the fastest modes of the learning curve are, in general, dominant initially, and the slower modes become dominant finally. The same is apparently true for the SGD algorithm (cf. (126)–(127)). This multimodal behavior complicates the problem of evaluating the time to convergence. Moreover, even for a unimodal learning curve (cf. (46)), time-to-convergence cannot be determined without knowledge of the initial and final excess MSE, $\varepsilon_*(0)$ and $\varepsilon_*(\infty)$, since this determines how many time-constants, say Q, are needed to satisfy the steady-state condition $[\varepsilon_*(0)-\varepsilon_*(\infty)]\,e^{-Q} \simeq \varepsilon_*(\infty)$. Specifically, if $\varepsilon_*(0) = Z\varepsilon_*(\infty)$, for some large number $Z \gg 1$, then it is required that $Q \cong \ln(Z)$ time constants. Consequently, analytical evaluation of time-to-convergence is a challenging problem that is beyond the scope of this paper. Nevertheless, the exact formulas for learning curves given in Sections 3.B and 3.C can be used to determine time-to-convergence graphically, or with trial-and-error solution of

$$\varepsilon_*(iK) = (1+\varepsilon)\varepsilon_*(\infty),$$

for some appropriate tolerance, ε (e.g., $\varepsilon = 1$ for 3 dB tolerance), using the exact formulas, (126), or (113) and (108), for $\varepsilon_*(iK)$.

References

[1] D.T.L. Lee, M. Morf and B. Friedlander, "Recursive least squares ladder estimation algorithms", *IEEE Trans. on Circuits and Sys.*, Vol. CAS-28, June 1981, pp. 467–481.

[2] A. Gersho, "Adaptive equalization of highly dispersive channels for data transmission", *The Bell Sys. Tech. J.*, Vol. 48, Jan. 1969, pp. 55–70.

[3] G. Ungerboeck, "Theory on the speed of convergence in adaptive equalizers for digital communication", *IBM J. Res. Develop.*, Nov. 1972, pp. 546–555.

[4] R.D. Gitlin, J.E. Mazo and M.G. Taylor, "On the design of gradient algorithms for digitally implemented adaptive filters", *IEEE Trans. on Circuit Theory*, Vol. CT-20, March 1973, pp. 125–136.

[5] B. Widrow, J.R. Glover, Jr., J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, E. Dong, Jr. and R.C. Goodlin, "Adaptive noise cancelling: Principles and applications", *Proc. IEEE*, Vol. 63, Dec. 1975, pp. 1692–1716.

[6] B. Widrow, J.M. McCool, M.G. Larimore and C.R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filter", *Proc. IEEE*, Vol. 64, Aug. 1976, pp. 1151–1162.

[7] R. D. Gitlin and S.B. Weinstein, "The effects of large interference on the tracking capability of digitally implemented echo cancellers", *IEEE Trans. on Commun.*, Vol. COM-26, June 1978, pp. 833–839.

[8] J. Zeidler, "Adaptive enhancement of multiple sinusoids in uncorrelated noise", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP-26, June 1978, pp. 240–254.

[9] R.D. Gitlin and S. B. Weinstein, "On the required tap-weight precision for digitally implemented, adaptive, mean-squared equalizers", *The Bell Sys. Tech. J.*, Vol. 58, February 1979, pp. 301–321.

[10] J.E. Mazo, "On the independence theory of equalizer convergence", *The Bell Sys. Tech. J.*, Vol. 58, May–June 1979, pp. 963–993.

[11] D.C. Farden, "Stochastic approximation with correlated data", *IEEE Trans. on Information Theory*, Vol. IT-27, Jan. 1981, pp. 105–113.

[12] D.C. Farden, "Tracking properties of adaptive signal processing algorithms", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP-29, June 1981, pp. 439–446.

[13] S.K. Jones, R.K. Cavin, III and W.M. Reed, "Analysis of error-gradient adaptive linear estimators for a class of stationary dependent processes", *IEEE Trans. on Information Theory*, Vol. IT-28, March 1982, pp. 318–329.

[14] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

[15] J.T. Rickard and J.R. Zeidler, "Second-order output statistics of the adaptive line enhancer", *IEEE Trans. on Acoustics, Speech, Signal Processing*, Vol. ASSP-27, Feb. 1979, pp. 31–39.

[16] N.J. Bershad, P.L. Feintuch, F.A. Reed and B. Fisher, "Tracking characteristics of the LMS adaptive line enhancer-response to a linear chirp signal in noise", *IEEE Trans. on Acoustics, Speech, Signal Processing*, Vol. ASSP-28, Oct. 1980, pp. 504–515.

[17] R.A. Monzingo and T.W. Miller, *Introduction to Adaptive Arrays*, John Wiley & Sons, New York, 1980.

[18] P. Monsen, "Linear estimation in an unknown quasi-stationary environment", *IEEE Trans. Sys., Man, Cybernetics*, Vol. SMC-1, July 1971, pp. 216–222.

[19] K.D. Senne, "Adaptive linear discrete-time estimation", Stanford University Center for Systems Research, Tech. Report 6778-5, SU-SEL-68-090, June 1968.

[20] L.L. Horowitz and K.D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays", *IEEE Trans. on Circuits and Systems*, Vol. CAS-28, June 1981, pp. 562–576.

[21] U. Grenander and G. Szegö, *Toeplitz Forms and Their Applications*, Berkeley, Calif.: Univ. of California Press, 1958.

[22] R.M. Gray, "On the asymptotic eigenvalue distribution of Toeplitz matrices", *IEEE Transactions on Information Theory*, Vol. IT-18, Nov. 1972, pp. 725–730.

[23] R.M. Gray, "Toeplitz and circulant matrices: II", Information Systems Laboratory Tech. Rept. No. 6504-1, April 1977, Stanford Univ. Center for Systems Research, Stanford, Calif. 94305.