A Family of Normalized LMS Algorithms *

Scott C. Douglas †
Department of Electrical Engineering
University of Utah
Salt Lake City, Utah 84112

Abstract– **In this letter, a derivation of the normalized LMS algorithm is generalized, resulting in a family of projection-like algorithms based upon an L_p-minimized filter coefficient change. The resulting algorithms include the simplified NLMS algorithm of Nagumo and Noda and an even simpler single-coefficient update algorithm based upon the maximum absolute value datum of the input data vector. A complete derivation of the algorithm family is given, and simulations are performed to show the convergence behaviors of the algorithms.**

IEEE SIGNAL PROCESSING LETTERS

vol. SPL-1, no. 3, pp. 49-51, March 1994.

EDICS Category No. SPL.4.5

*This research was supported in part by SRI International.

†Please address correspondence to: Scott C. Douglas, Department of Electrical Engineering, 2272 Merrill Engineering Building, University of Utah, Salt Lake City, UT 84112. (801) 581-4445. FAX: (801) 581-5281. Electronic mail address: douglas@ee.utah.edu.

1 Summary

The normalized least-mean-square (NLMS) algorithm, also known as the projection algorithm [1], is a useful method for adapting the coefficients of a finite-impulse-response (FIR) filter for a number of signal processing and control applications. The NLMS update is given by

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu \frac{e_k \mathbf{X}_k}{\sum_{m=1}^{L} x_{m,k}^2}, \tag{1}$$

where $\mathbf{W}_k = [w_{1,k} \ \cdots w_{L,k}]^T$ are the coefficients of the adaptive filter at time k, $\mathbf{X}_k = [x_{1,k} \ \cdots x_{L,k}]^T$ are the L samples of the input data in filter memory at time k, $e_k = d_k - \mathbf{W}_k^T \mathbf{X}_k$ is the error between the adaptive filter output and the desired signal d_k, and μ is a user-specified convergence parameter. This algorithm has two distinct advantages over the least-mean-square (LMS) algorithm: 1) potentially-faster convergence speeds for both correlated and whitened input data [2, 3, 4], and 2) stable behavior for a known range of parameter values ($0 < \mu < 2$), independent of the input data correlation statistics [1, 2]. The NLMS algorithm requires a minimum of one additional multiply, divide and addition over the LMS algorithm to implement for shift-input data. Even so, the multiplies required for the algorithm update may still be prohibitive in certain high-data-rate applications. In these situations, it is useful to determine modified versions of the NLMS algorithm that retain the fast convergence properties of the algorithm while reducing the amount of computation per iteration. One such modified algorithm, first suggested by Nagumo and Noda [2], is

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu \frac{e_k \mathrm{sgn}(\mathbf{X}_k)}{\sum_{m=1}^{L} |x_{m,k}|}. \tag{2}$$

This update is similar to that in (1) but allows nonlinear modification of the data vector elements.

In this letter, we derive a generalized class of normalized LMS algorithms of the form

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu e_k F_q(\mathbf{X}_k) \tag{3}$$

$$[F_q(\mathbf{X}_k)]_i = \begin{cases} \dfrac{|x_{i,k}|^{q-1}\mathrm{sgn}(x_{i,k})}{\sum_{m=1}^{L} |x_{m,k}|^q} & \text{if } 1 \leq q < \infty \\ \dfrac{1}{x_{n,k}}\delta_{i-n} & \text{if } q = \infty, \end{cases} \tag{4}$$

where $[F_q(\cdot)]_i$ denotes the ith element of the vector-valued function $F_q(\cdot)$, δ_j is the Kronnecker delta function, and n is any one integer for which $|x_{n,k}| = \max_{1 \leq j \leq L} |x_{j,k}|$. For $q = 2$, this update reduces to that of the NLMS algorithm in (1), and for $q = 1$, this update reduces to the algorithm of Nagumo and Noda given in (2). We provide a theoretical derivation of these algoriths showing that for $\mu = 1$, the algorithm in (3)–(4) for any valid q is the solution to the following optimization problem:

$$\text{minimize} \qquad ||\mathbf{W}_{k+1} - \mathbf{W}_k||_p \tag{5}$$

$$\text{subject to} \qquad d_k - \mathbf{W}_{k+1}^T \mathbf{X}_k = 0, \tag{6}$$

where $|| \cdot ||_p$ denotes the L_p norm and p is determined from the equation $1/p + 1/q = 1$. Thus, the adaptive algorithm update in (3)–(4) provides the minimum change in an L_p-norm sense of the weights to exactly satisfy the filtering relationship between the input data and the desired response at time k, similar to a projection in the L_2-norm case.

Examining the algorithm in (3)–(4) for $q = \infty$, we discover a simple but powerful adaptation algorithm given by

$$w_{i,k+1} = \begin{cases} w_{i,k} + \mu \dfrac{e_k}{x_{i,k}}, & \text{if } |x_{i,k}| = \max\limits_{1 \leq j \leq L} |x_{j,k}| \\ w_{i,k}, & \text{otherwise.} \end{cases} \tag{7}$$

In this expression of the update, the maximum absolute data value $|x_{i,k}| = \max_{1 \leq j \leq L} |x_{j,k}|$ is assumed to be unique; if not, a single filter coefficient from the set $\{w_{i,k} : |x_{i,k}| = \max_{1 \leq j \leq L} |x_{j,k}|\}$ is chosen randomly for updating. Thus, the only filter coefficient updated at time k is a coefficient associated with an input sample which has the largest absolute value of all input data samples currently in filter memory. This algorithm requires a search through the input data vector elements but only requires one multiply, one divide, and one addition per iteration, simplifying its implementation in hardware. Efficient methods for maintaining the maximum data element across a shift-input data window exist [5], and a divide-and-conquer strategy requires at most $\log_2 L$ compares at each iteration. Thus, the new algorithm may prove useful in situations where excessive filter lengths preclude updating every weight at each iteration.

2 Derivation

We now show that the family of NLMS algorithms described by (3)–(4) solves the optimization problem in (5)–(6). Our derivation follows a similar derivation presented in [6] for the Nagumo and Noda algorithm and uses the following theorem.

Theorem: Let $\mathbf{A}$ be a nonzero vector contained in the vector space $\mathcal{R}^L$, and let b be a scalar quantity. Then, the minimum L_p-norm solution vector $\mathbf{Z}$ to a consistent linear equation $\mathbf{A}^T\mathbf{Z} = b$ is given by

$$\mathbf{Z} = bF_q(\mathbf{A}), \tag{8}$$

where the vector function $F_q(\cdot)$ is given by (4).

Proof:

Let a_i and z_i denote the ith elements of the vectors $\mathbf{A}$ and $\mathbf{Z}$, respectively. Then,

$$\begin{aligned} |b| &= \left|\sum_{i=1}^{L} a_i z_i\right| && (9) \\ &\leq ||\mathbf{Z}||_p ||\mathbf{A}||_q && (10) \end{aligned}$$

where (10) follows from (9) from the Hölder inequality with $1/p + 1/q = 1$. Thus, for the nonzero vector $\mathbf{A}$, we have

$$||\mathbf{Z}||_p \geq \frac{|b|}{||\mathbf{A}||_q}. \tag{11}$$

Consequently, the following inequality holds:

$$\min_{\mathbf{A}^T\mathbf{Z}=b} ||\mathbf{Z}||_p \geq \frac{|b|}{||\mathbf{A}||_q}. \tag{12}$$

Let $\overline{\mathbf{Z}}$ be a solution vector to the equation $\mathbf{A}^T\mathbf{Z} = b$. Note that $\overline{\mathbf{Z}}$ is not unique, but that it satisfies

$$||\overline{\mathbf{Z}}||_p \geq \min_{\mathbf{A}^T\mathbf{Z}=b} ||\mathbf{Z}||_p \tag{13}$$

for all $\overline{\mathbf{Z}}$ in $\mathcal{R}^L$. Now, let

$$\overline{\mathbf{Z}} = bF_q(\mathbf{A}). \tag{14}$$

It can be seen that, for $1 \leq q < \infty$,

$$\begin{aligned} ||\overline{\mathbf{Z}}||_p &= |b| \left(\frac{\sum_{i=1}^{L} |a_i|^{p(q-1)}}{||\mathbf{A}||_q^{pq}} \right)^{1/p} && (15) \\ &= \frac{|b|}{||\mathbf{A}||_q} \left(\frac{\sum_{i=1}^{L} |a_i|^{p(q-1)}}{||\mathbf{A}||_q^{p(q-1)}} \right)^{1/p}. && (16) \end{aligned}$$

Using the relationship $p = q/(q-1)$, the term inside the parentheses of (16) can be shown to be equal to one. Thus, from (12) and (16), we have

$$||\overline{\mathbf{Z}}||_p = \min_{\mathbf{A}^T\mathbf{Z}=b} ||\mathbf{Z}||_p. \tag{17}$$

Considering the case $(p = 1, q = \infty)$, we find from (12) and (14) that

$$||\overline{\mathbf{Z}}||_1 = \frac{b}{||\mathbf{A}||_\infty} = \min_{\mathbf{A}^T\mathbf{Z}=b} ||\mathbf{Z}||_1. \tag{18}$$

Therefore, (8) follows. □.

To see how the theorem enables the solution to the problem posed in (5)–(6), assign $\mathbf{Z} = \mathbf{W}_{k+1} - \mathbf{W}_k$, $\mathbf{A} = \mathbf{X}_k$, and $b = e_k$. Then, from the definition of the error e_k, we have

$$e_k = d_k - \mathbf{X}_k^T\mathbf{W}_k \tag{19}$$

$$= (d_k - \mathbf{X}_k^T\mathbf{W}_{k+1}) + \mathbf{X}_k^T(\mathbf{W}_{k+1} - \mathbf{W}_k). \tag{20}$$

If the constraint in (6) is satisfied, then from our assignments of $\mathbf{Z}$, $\mathbf{A}$, and b,

$$\mathbf{X}_k^T(\mathbf{W}_{k+1} - \mathbf{W}_k) = e_k \quad \rightarrow \quad \mathbf{A}^T\mathbf{Z} = b, \tag{21}$$

and thus the optimization problem in (5)–(6) is the same as the minimization of $||\mathbf{Z}||_p$ subject to $\mathbf{A}^T\mathbf{Z} = b$. Therefore, from (17), the optimum update for $\mathbf{W}_k$ is given by (3)–(4).

3 Simulations

We now present simulations of the simplified update algorithm to compare its performance with the standard NLMS and Nagumo and Noda algorithms for a six-coefficient FIR system identification task. The input data for this system was generated as

$$x_{i,k} = \sin\left(\frac{2\pi(k-i+1)}{15}\right) + v_{k-i+1}, \tag{22}$$

where v_k is a white Gaussian data sequence with variance $\sigma_v^2 = 0.01$. The output of the system to be identified was generated by filtering this input using a six-tap filter with unity coefficients and adding white Gaussian noise with variance $\sigma_n^2 = 0.01$ to each sample. Step sizes for the three algorithms were chosen by trial-and-error such that each algorithm produced the same average excess mean-square error at convergence. The initial coefficients $\mathbf{W}_0$ were found by perturbing

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.