A PERFORMANCE-WEIGHTED MIXTURE OF LMS FILTERS

Suleyman S. Kozat

Koc University
Istanbul, Turkey
Email: skozat@ku.edu.tr

Andrew C. Singer

University of Illinois at Urbana Champaign
Urbana, IL 61801, USA
Email: acsinger@uiuc.edu

ABSTRACT

In this paper, we explore the use of a particular multi-stage adaptation algorithm for a variety of adaptive filtering applications where the structure of the underlying process to be estimated is unknown. The proposed algorithm uses a performance-weighted mixture of LMS filters of various orders to construct its final output. The algorithm is analyzed in a stochastic context with respect to its convergence and mean-square error (MSE) behaviors and is shown to achieve the best MSE performance of the constituent algorithms in the mixture. Through simulations, it has been observed that the mixture structure can offer considerable performance improvement for both stationary and time varying observation sequences.

Index Terms— Universal, prediction, model combination, model mixture, LMS

I. INTRODUCTION

In this paper, we investigate a particular combination of adaptive filters, each running the LMS algorithm for adaptation, as introduced in [1]. The adaptive filters are combined based on their performance on the underlying task. Here, we consider one-step-ahead prediction, however, the algorithm can be readily extended to other applications.

While the minimum mean-square error (MMSE) optimal filtering structure for a wide-sense stationary random process contains a single filter, the number of filter parameters (if not the optimal values of these parameters) are usually assumed unknown *a priori*. In fact, for nonstationary observation sequences, the number and values of these optimal parameters will generally be time varying. Furthermore, even when the number of parameters and the adaptive algorithm to be used for training these parameters are known, it still remains difficult to fix the appropriate parameters for the selected adaptation algorithm [2]. For example, combining two LMS algorithms one with comparably larger adaptation coefficient and hence quick convergence, and the other with comparably smaller adaptation coefficient and hence better steady state performance, can produce an algorithm that will have the benefits of each, depending on the state of the underlying process. Thus, fixing a specific filter structure (or adaptation algorithm) has potentially significant drawbacks due to the lack of *a priori* information about the observation sequence. A mixture algorithm attempts to overcome these problems by combining multiple candidate filter structures or adaptation algorithms, with the goal of sequentially achieving the performance of the best among them.

In this paper, we investigate the LMS-Bayesian algorithm as shown in Fig. 1. The LMS Bayesian algorithm was introduced in [1] and a simplified analysis of its convergence behavior was given in [3]. Here, we will extend this analysis and remove most of the assumptions used in the derivations and also demonstrate the performance of the LMS-Bayesian algorithm under different scenarios. The LMS-Bayesian algorithm consists of multiple linear predictors of orders 1 to m in the first stage, operating in parallel on the observation sequence. The predictors are each updated with the LMS algorithm, and their predictions are combined by a performance-weighted mixture. Instead of selecting a particular predictor order, the LMS-Bayesian algorithm adaptively combines the output of several different predictors of varying order to achieve the performance of the best predictor given any realization. The weight of each predictor in the final mixture is updated based on its performance on the past observations. This way, we exploit the time-dependent nature of the best choice from the constituent algorithms given any realization. By decoupling the constituent algorithms from one another and from the mixture stage, the algorithm can take advantage of regions in the data where different constituent models outperform the others. Furthermore, the constituent predictors using the LMS algorithm of orders from 1 to m can be efficiently calculated by using lattice filters. A lattice implementation of an mth order LMS filter will produce predictions of all the lower-order linear predictors; orders from 1 to $m-1$ with the computational complexity $O(m)$. Hence, the mixture structure can be implemented with a complexity similar to tapped-delay-line implementation of the LMS algorithm. For a one-step-ahead prediction problem, an explicit description of the LMS-Bayesian algorithm is given as follows. Let $\hat{x}_k(n)$ be the output of a sequential linear predictor as obtained by the LMS algorithm with model order k, i.e.,

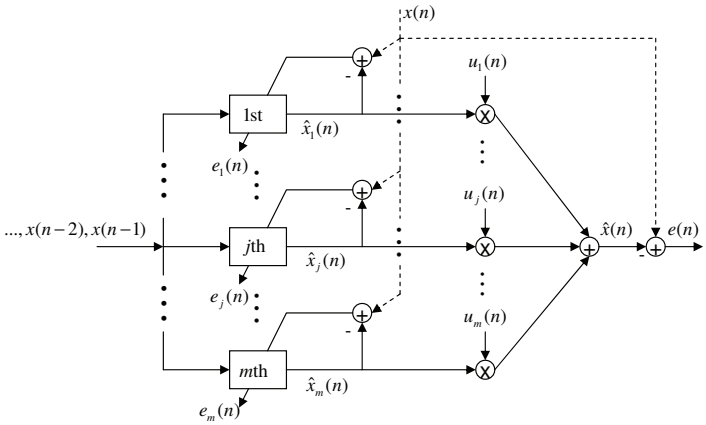

Fig. 1. LMS-Bayesian Algorithm.

$$\begin{aligned}\hat{x}_k(n) &= \boldsymbol{w}_k^T(n-1)\boldsymbol{x}_k(n-1),\\ e_k(n) &= x(n)-\hat{x}_k(n),\\ \boldsymbol{w}_k(n) &= \boldsymbol{w}_k(n-1)+\mu e_k(n)\boldsymbol{x}_k(n-1), \qquad (1)\end{aligned}$$

where μ is a constant to control stability and rate of convergence, and $e_k(n)$ is the prediction error to be minimized in the mean-square. The weight and input vectors are given by $\boldsymbol{w}_k(n-1) = [w_{1,k}^{n-1},\ldots,w_{k,k}^{n-1}]^T$ and $\boldsymbol{x}_k(n-1) = [x(n-1),\ldots,x(n-k)]^T$, respectively. Define the LMS-Bayesian predictor as the following weighted sum over linear predictors of order less than or equal to m:

$$\begin{aligned}\hat{x}(n) &= \sum_{k=1}^{m} u_k(n)\hat{x}_k(n), \qquad (2)\\ u_k(n) &= \frac{\exp[-c\, l_{n-1}(x,\hat{x}_k)]}{\sum_{j=1}^{m}\exp(-c\, l_{n-1}(x,\hat{x}_j))},\end{aligned}$$

where c is a positive constant and $u_k(n)$, the mixture weights, are proportional to the performance of the kth-order predictor on the data observed so far. The performance, $l_{n-1}(x,\hat{x}_k) = \sum_{i=1}^{n-1}[x(i)-\hat{x}_k(i)]^2$, is the accumulated squared prediction error that results from $\hat{x}_k(n)$. For each new sample at time n, these coefficients are updated using (1) and (2).

The organization of this paper is as follows. In Section II, we provide the a prior art. In Section III, we present the main convergence results of this paper. We then provide an outline of the proofs, due to space limitations. We then illustrate the performance gains achieved by the LMS-Bayesian algorithm for both stationary and non-stationary observations.

II. BACKGROUND

The LMS-Bayesian predictor is similar to a certain universal linear predictor introduced in [4] wherein the authors use the same performance-weighted mixture to combine predictions of multiple-order linear predictors, each using the RLS algorithm instead of the LMS algorithm. Although the universal linear predictor is shown to asymptotically achieve the performance of the best constituent predictor for any bounded but otherwise deterministic sequence in [4], we analyze the LMS-Bayesian algorithm in a stochastic environment and provide convergence results for its mean-square error and internal weights. The results for deterministic data do not hold in a fairly general stochastic context; for example, if the observation sequence is a stationary Gaussian process, the boundedness assumption is invalidated.

In [2], the authors introduced a convex combination of two adaptive filters and investigated the mean-square convergence properties in a system identification framework. The coefficient of the convex combination is determined by means of a stochastic gradient algorithm in order to minimize the error of the overall structure. For the prediction problem considered in this paper (and with the same notation used in this paper), the recursion for the parameter of the convex combination $\lambda(n) = u_1(n)$ (and naturally $1-\lambda(n) = 1-u_1(n) = u_2(n)$) is given by

$$u_1(n) = \frac{1}{1+\exp[-a(n)]} \qquad (3)$$

where $a(n)$ is updated as

$$\begin{aligned}&a(n+1) = a(n) + \mu_a[\hat{x}_1(n)-\hat{x}_2(n)]\\ &\{x(n)-u_1(n)\hat{x}_1(n)-[1-u_1(n)]\hat{x}_2(n)\}u_1(n)[1-u_1(n)],\end{aligned}$$

where μ_a is the learning rate. The parameter $a(n)$ defines the combination parameter via the sigmoid nonlinearity and is used to minimize the overall error. The sigmoid is used to constrain to $u_1(n)$ to the range $[0,1]$ and also to minimize the gradient noise in adaptation. Using the update for $a(n+1)$ in (3), the update for $u_1(n)$ is given by

$$u_1(n+1) = \frac{1}{1+A(n)}, \qquad (4)$$

where

$$\begin{aligned}A(n) \triangleq \frac{[1-u_1(n)]}{u_1(n)}\exp\Big(&-\mu_a[\hat{x}_1(n)-\hat{x}_2(n)]\\ &\{x(n)-[u_1(n)\hat{x}_1(n)+u_2(n)\hat{x}_2(n)]\}\,u_1(n)[1-u_1(n)]\Big).\end{aligned}$$

For a mixture of only two algorithms, the combination weight for the LMS-Bayesian algorithm introduced in (2) is given by [5]

$$u_1(n+1) = \frac{1}{1+B(n)}, \qquad (5)$$

where

$$\begin{aligned}B(n) \triangleq \frac{[1-u_1(n)]}{u_1(n)}\exp\Big(&-2c[\hat{x}_1(n)-\hat{x}_2(n)]\\ &\{x(n)-[\hat{x}_1(n)+\hat{x}_2(n)]/2\}\Big)\end{aligned}$$

The recursion in (4) is similar to (5). The main difference is the term $u_1(n)[1-u_1(n)]$ in the exponent of (4), which is not present in (5). This multiplicative term will avoid or dampen the convergence of weight coefficients when either of the weight coefficients are near 0, i.e., near convergence. To remedy this, in [2], the authors introduce an ad-hoc update to constrain either the weight $u_1(n)$ or $a(n)$ to avoid the boundaries.

III. CONVERGENCE RESULTS FOR THE LMS-BAYESIAN ALGORITHM

In this section, we will investigate the convergence behavior of the LMS-Bayesian algorithm for stationary Gaussian data in a stochastic context. We will demonstrate that the LMS-Bayesian algorithm is consistent in a certain sense for variance-ergodic Gaussian random processes (i.e., where the variance estimated from sample paths converges to the true variance). The results hold for more general stationary Gaussian processes, however, we use variance ergodicity to simplify the presentation. The main results of the paper are given by the following theorem.

Theorem 1: *For a wide-sense stationary, variance-ergodic, Gaussian observation process, the mixture coefficients of the LMS-Bayesian algorithm $u_p(n)$ are consistent with probability 1, such that*

$$u_p(n) \to 0, \text{ as } n \to \infty \text{ (pr)}, \; p \neq \min(w, m), \qquad (6)$$

where w is the order of the MMSE-optimal linear filter of order less than or equal to m and when the learning parameter μ for the LMS algorithms in the mixture is selected from a nontrivial interval.

For example, if the underlying process $x(n) = \sum_{k=1}^{w} c_k x(n-k) + \varepsilon(n)$, where $\varepsilon(n)$ is a sequence of i.i.d. Gaussian random variables with zero mean and variance σ_ε^2, then w is the order of this auto-regressive process. For a general stationary Gaussian process, the required order can be arbitrarily large.

From Theorem 1, we further conclude the following.

Corollary: *For a wide-sense stationary, mean ergodic, Gaussian observation process, the mixture coefficients of the LMS-Bayesian algorithm $u_p(n)$ are consistent in mean and mean square, such that*

$$E[u_p(n)] \to 0, \text{ as } n \to \infty, \; p \neq \min(w, m),$$
$$E[u_p^2(n)] \to 0, \text{ as } n \to \infty, \; p \neq \min(w, m), \qquad (7)$$

where w is the order of the MMSE-optimal linear filter of order less than or equal to m and when the learning parameter μ for the LMS algorithms in the mixture is be selected from a nontrivial interval.

The proof of Corollary is due to the boundness of $u_p(n)$, i.e., $0 \leq u_p(n) \leq 1$. Using this corollary, we give the following theorem.

Theorem 2: *For a wide-sense stationary, mean ergodic, Gaussian observation process*

$$\lim_{n\to\infty} E\left\{[x(n) - \hat{x}(n)]^2\right\} \leq \min_{p=1,\ldots,m} \lim_{n\to\infty} E\left\{[x(n) - \hat{x}_p(n)]^2\right\},$$

when the learning parameter μ for the LMS algorithms in the mixture is selected from a nontrivial interval.

This result implies that the LMS-Bayesian algorithm does not asymptotically do worse than the best of the mixture algorithms in terms of the final MSE.

Outline of Proof of Theorem 1: We observe that, for all p, from the definition in Equation (2), the weight coefficients $u_p(n)$ can be expressed as

$$u_p(n) = \frac{1}{\sum_{k=1}^{m} \exp\left(c \sum_{l=1}^{n} [e_p^2(l) - e_k^2(l)]\right)} = \frac{1}{\sum_{k=1}^{m} \exp\left[c\, Y_{p,k}(n)\right]}, \qquad (8)$$

where we define $Y_{p,k}(n) \stackrel{\triangle}{=} \sum_{l=1}^{n} [e_p^2(l) - e_k^2(l)]$ for any p and k. We next investigate the behavior of the $Y_{p,k}(n)$'s and derive the convergence results based on $E[Y_{p,k}(n)]$.

By the Chebyshev inequality, for any $\epsilon \in R^+$,

$$\Pr\left[\left|\frac{Y_{p,k}(n) - E[Y_{p,k}(n)]}{n}\right| < \epsilon\right] > 1 - \frac{\text{Var}[Y_{p,k}(n)]}{n^2\epsilon^2}, \qquad (9)$$

where $\text{Var}(x)$ is the variance of x. We state without proof the following Lemma,

Lemma: $\text{Var}[Y_{p,k}(n)] = o(n^2)$, i.e., $\lim_{n\to\infty} \frac{\text{Var}[Y_{p,k}(n)]}{n^2} = 0$.

Hence, in a certain sense, $Y_{p,k}(n)$ behaves like $E[Y_{p,k}(n)]$ asymptotically and in this sense $E[Y_{p,k}(n)]$ can be used in (8) to investigate the asymptotic behavior of $u_p(n)$. Since the behavior of each $E[e_p^2(l)]$ (or $E[e_k^2(l)]$) is well studied [6], the behavior of

$$E[Y_{p,k}(n)] = E\{\sum_{l=1}^{n} [e_p^2(l) - e_k^2(l)]\} = \sum_{l=1}^{n} \{E[e_p^2(l)] - E[e_k^2(l)]\}$$

can also be derived following similar lines. We next show that for predictors with orders $p \neq w$, $E[Y_{p,k}(n)]$ diverges for at least one of the terms in (8), which causes $u_p(n)$ to vanish. This completes the outline of the proof. □

IV. SIMULATIONS

In this section, we illustrate the performance of the LMS-Bayesian algorithm for stationary and nonstationary data. The first set of experiments involve prediction of a 4th-order AR process generated by $x(n) = 0.9x(n-1) - 0.6x(n-2) + 0.5x(n-3) - 0.3x(n-4) + w(n)$, where $w(n)$ is a sample function of a stationary white Gaussian process with mean zero and variance 0.1. As the constituent algorithms, we use one-step-ahead predictors with orders from 1 to 10, all using the LMS algorithm for adaptation. The adaptation coefficient μ for the LMS algorithms are set to 0.15. The LMS-Bayesian algorithm uses a soft performance-weighted combination of the constituent algorithms instead of making hard decision at each sample time. Hence, we compare its performance to an algorithm (named as "pick" in the figures) that picks the output of the best performing model (up to this time) and repeat the prediction, i.e., pick $\hat{x}_k(n)$ if $k = \arg\min_i \{\sum_{t=1}^{n-1} [x(t) - \hat{x}_i(t)]^2\}$. We also use a modified version of this algorithm based on the well known MDL criteria and choose the best algorithm as, pick $\hat{x}_k(n)$ if $k = \arg\min_i \left\{\sum_{t=1}^{n-1} [x(t) - \hat{x}_i(t)]^2 + (p/2)\log(n-1)\right\}$, where the $(p/2)\log(n)$ term is included to penalize higher model orders. In the this set of experiments, we do not compare the performance of the LMS-Bayesian algorithm

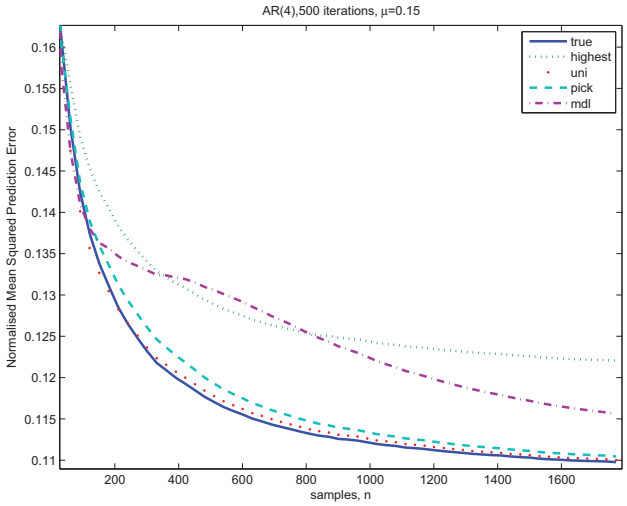

Fig. 2. Running average prediction results for the 4th-order autoregressive process. The correct order "true", i.e., 4th-order; the largest-order "highest", i.e., 10th order; the LMS-Bayesian indicated by red dotted line "uni"; the "pick", i.e., pick $\hat{x}_k(n)$ if $k = \arg\min_i \sum_{t=1}^{n-1}(x(t)-\hat{x}_i(t))^2$; the "mdl", i.e., pick $\hat{x}_k(n)$ if $k = \arg\min_i \sum_{t=1}^{n-1}(x(t)-\hat{x}_i(t))^2 + (p/2)\log(n)$.

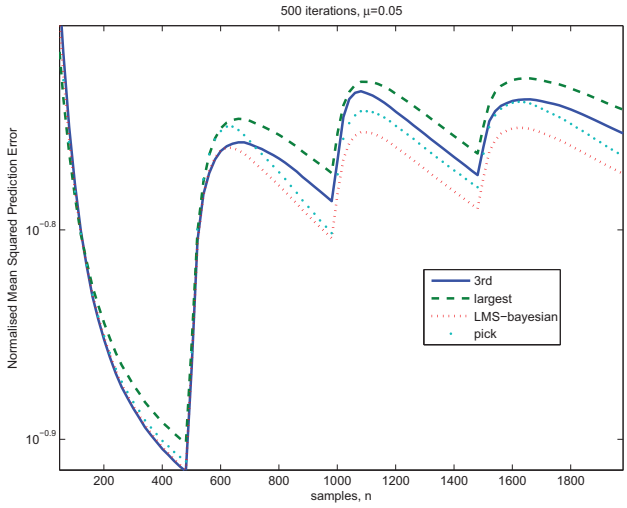

Fig. 3. Running average prediction results nonstationary data. The largest-order "largest", i.e., 10th order; the 3rd-order "3rd"; the LMS-Bayesian algorithm "LMS-bayesian"; "pick", i.e., pick $\hat{x}_k(n)$ if $k = \arg\min_i \sum_{t=1}^{n-1}(x(t)-\hat{x}_i(t))^2$.

with the convex combination algorithm since it would be unfair to the convex combination, which uses only two algorithms. However we observe that an LMS-Bayesian algorithm using only two filters as the convex combination performs similarly to the convex combination algorithm [5]. Hence, the LMS-Bayesian algorithm can also be used as a generalization of the convex combination algorithm for more than two filters. In Fig. 1, we plot the normalized running average prediction error for each of the algorithms, as well as the performance of the correct order LMS predictor (4th-order), and the largest order LMS predictors (10th-order). The performance of the LMS-Bayesian algorithm is superior to the performance of the other algorithms. At the start of the experiment, since we plot the sequential performance of the algorithms, the lower-order LMS algorithms outperform the larger-order ones. This is due to the faster convergence of the lower order models since they have fewer parameters. However, as time progresses, the performance of the larger order LMS algorithms improves.

The performance of the LMS-Bayesian algorithm for nonstationary data is shown in Fig. 3, where it is applied to an autoregressive process that switches between a 2nd-order and a 4th-order process every 500 samples, i.e., $x(n) = -1.4x(n-1)-0.74x(n-2)+\epsilon(n)$ and $x[n] = 0.9x(n-1)-0.25x(n-2)-0.1x(n-3)-0.2x(n-4)+\epsilon(n)$ with $\sigma^2 = 0.1$ and $\mu = 0.05$. The signal starts as a 2nd-order process and then switches back and forth as a 4th- and 2nd-order process at time sample 500, 1000, and 1500. The adaptation parameters are not selected to optimize the convergence. For the algorithms presented in Fig. 3, we also calculated the total squared prediction error for an effective window of size 100 samples for each algorithm due to nonstationarity of the data and calculated the mixture weights based on the performance in this sliding window.

V. CONCLUSION

In this paper, we investigated a particular multistage adaptive filter algorithm. The LMS-Bayesian algorithm is analyzed in terms of its MSE and mean convergence characteristics. With the aid of some simplifying assumptions, the LMS-Bayesian algorithm is shown to converge to the final MSE of the best predictor used in the constituent class.

VI. REFERENCES

[1] S. S. Kozat and A. C. Singer, "Multi-stage adaptive signal processing algorithms," in *Proc. Sensor Array and Multi-Channel Process.*, 2000, pp. 380-384.

[2] J. A.-Garcia, A. R. F.-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54, 1078-1090, March 2006.

[3] S. S. Kozat and A. C. Singer, "Further results in multistage adaptive filtering," in *Proc. ICASSP*, 2002, pp. 1329-1332.

[4] A. C. Singer and M. Feder, "Universal linear prediction by model order weighting," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2685 - 2699, Oct. 1999.

[5] S. S. Kozat and A. C. Singer, "A performance weighted combination of LMS filters," to be submitted.

[6] M. Honig and D.G. Messerschmitt, *Adaptive Filters: Structures, Algorithms and Applications*, Springer, 1984.