

Universal Linear Prediction by Model Order Weighting

Andrew C. Singer, *Member, IEEE*, and Meir Feder, *Fellow, IEEE*

Abstract—A common problem that arises in adaptive filtering, autoregressive modeling, or linear prediction is the selection of an appropriate order for the underlying linear parametric model. We address this problem for linear prediction, but instead of fixing a specific model order, we develop a sequential prediction algorithm whose *sequentially accumulated average squared prediction error* for any bounded individual sequence is as good as the performance attainable by the best sequential linear predictor of order less than some M . This predictor is found by transforming linear prediction into a problem analogous to the sequential probability assignment problem from universal coding theory. The resulting universal predictor uses essentially a performance-weighted average of all predictors for model orders less than M . Efficient lattice filters are used to generate the predictions of all the models recursively, resulting in a complexity of the universal algorithm that is no larger than that of the largest model order. Examples of prediction performance are provided for autoregressive and speech data as well as an example of adaptive data equalization.

Index Terms—Adaptive filters, Bayes procedures, learning systems, least squares methods, model order, prediction methods, sequential decision procedures, universal coding, universal methods.

I. INTRODUCTION

Autoregressive (AR) modeling by predictive least-squares, or linear prediction, forms the basis of a wide variety of signal processing and communication systems including adaptive filtering and control, speech modeling and coding, adaptive channel equalization, parametric spectral estimation, and system identification. In linear prediction, the signal $x[t]$ at time t is modeled (or predicted) as a linear combination of, say, the previous p samples, i.e.,

$$x[t] \sim \hat{x}_p[t] = \sum_{k=1}^p c_k x[t-k].$$

To apply linear prediction to the data, either for prediction or for modeling purposes, we have to determine the value of

Manuscript received August 26, 1997; revised February 20, 1999. This work was prepared through collaborative participation in the Advanced Telecommunications/Information Distribution Research Program (ATIRP) Consortium and supported by the U.S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0002. It was also supported in part by a grant from the Israeli Science Foundation. The associate editor coordinating the review of this paper and approving it for publication was Dr. Ali H. Sayed.

A. C. Singer is with the Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL 61801 USA (e-mail: acsinger@uiuc.edu).

M. Feder is with the Department of Electrical Engineering-Systems, Tel-Aviv University, Tel-Aviv, Israel (e-mail: meir@eng.tau.ac.il).

Publisher Item Identifier S 1053-587X(99)07572-8.

the linear prediction coefficients c_k , $k = 1, \dots, p$ as well as the order p . Given a batch of data points $x^n = x[1], \dots, x[n]$ and a fixed order p , a common way to select the prediction coefficients is to minimize the total squared prediction error, i.e., select $C^n = [c_1^n, \dots, c_p^n]^T$ so that

$$C^n = \arg \min_{C=[c_1, \dots, c_p]} \sum_{t=1}^n \left(x[t] - \sum_{k=1}^p c_k x[t-k] \right)^2. \quad (1)$$

The resulting residual square error in batch fitting the p th-order linear prediction coefficients to the data is denoted

$$E_n(x, \hat{x}_p^B) = \sum_{t=1}^n \left(x[t] - \sum_{k=1}^p c_k^n x[t-k] \right)^2. \quad (2)$$

The selection of the model order p is an important, but often difficult, aspect of applying linear prediction to a particular application. Intuitively, an appropriate model order for a particular application depends both on the amount of memory in the process and on the length of data over which the model will be applied. On one hand, larger model orders can capture the dynamics of a richer class of signals. On the other hand, larger model orders also require proportionally larger data sets for the parameters to be accurately estimated.

Some of the methods of model order selection that are often used in practice include the Akaike information criterion (AIC) [1], the minimum description length (MDL) proposed by Rissanen [2], the Bayes information criterion (BIC) of Schwarz [3], which is equivalent to the MDL in many settings, and the predictive least-squares (PLS) principle of Rissanen [4], [5]. In their original form, the AIC and MDL criteria comprise an explicit balance between the likelihood of the data given the model and a penalty term for the complexity of the model. Intuitively, in MDL, the goal is to minimize the number of bits that would be required to “describe” the data. Since the data could be modeled parametrically and then block encoded, one approach would be to measure the block log-likelihood of the data given a model and then penalize this model by the additional number of bits required to encode its parameters. For example, for an AR process with white Gaussian noise drive, the log-likelihood of the data given the AR parameters is directly proportional to the total squared linear prediction error over the data. The leading term of the penalty that the MDL assigns to a model of order p for such a signal of length n is $(p/2) \log(n)$. Hence, for such a signal, according to the original definition of the MDL, the model order is basically

selected by finding the minimum of

$$\min_{c_1, \dots, c_p} \sum_{t=1}^n \left(x[t] - \sum_{k=1}^p c_k x[t-k] \right)^2 + \frac{p}{2} \log n = E_n(x, \hat{x}_p^B) + \frac{p}{2} \log n \quad (3)$$

with respect to p . Note that in many recent applications of the MDL (see e.g., [6]), a more refined penalty term is suggested.

The PLS criterion suggested later in [4] examines a sequential coding of the data, where the codelength of each data point is proportional to its square sequential prediction error. Since the parameters of the encoder are not optimized over the entire block of data, but rather are determined online, there is no “batch” penalty for their use. However, there is an implicit penalty since for higher order models, a larger squared prediction, or encoding, error is incurred due to the lack of sufficient data to accurately estimate the parameters. In fact, it was shown that in many cases, the PLS is essentially another version of the MDL principle as it leads to the same balance between likelihood and model complexity as the original MDL.

This issue needs a further explanation, as it is relevant to the main subject of the paper. The difference between the “sequential” error leading to the PLS and the “batch” prediction error is subtle and lies both in the method used to compute the predictor coefficients c_k , $k = 1, \dots, p$ and in the samples over which the error is computed. The least-squares *batch prediction error* $E_n(x, \hat{x}_p^B)$, which is defined above, is the total squared prediction error that results from applying the fixed set of predictor coefficients obtained by minimizing the square prediction error over the same set of data. In the notation above, \hat{x}_p^B is the *batch* predicted sequence. The *sequential prediction error*, on the other hand, is the accumulated squared prediction error that results from sequential application of a time-varying set of predictor coefficients $C^t = [c_1^t, \dots, c_p^t]^T$. A common way to obtain the coefficients C^{t-1} to predict $x[t]$ is to use the coefficients that minimize the batch error over the samples $x[1], \dots, x[t-1]$ observed so far, i.e., the coefficients attaining $E_{t-1}(x, \hat{x}_p^B)$. The resulting sequential prediction error is

$$l_n(x, \hat{x}_p) \triangleq \sum_{t=1}^n \left(x[t] - \sum_{k=1}^p c_k^{t-1} x[t-k] \right)^2. \quad (4)$$

Note that now, the linear prediction coefficients are optimized only over the data available up to but not including the value to be predicted. In this sense, the sequential prediction error is a “fair” measure of performance for prediction. The notation \hat{x}_p denotes the predicted sequence. In addition, note that the notations $E_n(\cdot, \cdot)$ and $l_n(\cdot, \cdot)$ for the accumulated square error adopted in this paper actually stand for the standard Euclidean norm of the batch and sequential prediction error signals.

By definition, for a given x^n , the batch error $E_n(x, \hat{x}_p^B)$ is a monotonic, nonincreasing function of p since the class of models of order p contains all models of order less than p . This is not true for the sequential error $l_n(x, \hat{x}_p)$. In fact, lower

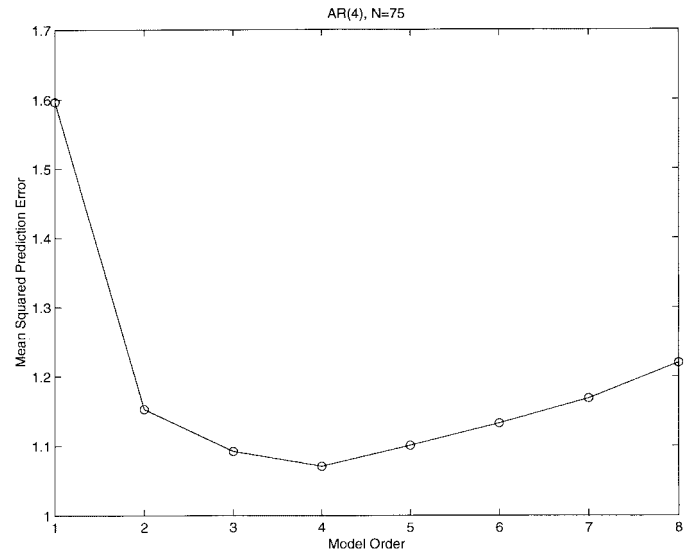


Fig. 1. Sequentially accumulated average squared prediction errors of a fourth-order autoregressive process shown for linear predictors of order $1, \dots, 8$.

order models can outperform larger order models, i.e., have a smaller sequential prediction error. This is best visualized by an example. Consider a fourth-order AR process, and suppose that 75 samples of the process have been observed. An estimate of the parameters of any order larger than four will concentrate around the true parameters. However, the fourth-order estimate will have a lower variance than would, say, a seventh-order estimate. Therefore, although these parameter estimates will asymptotically coincide, every time in which the seventh-order model is used to predict the next sample, the prediction error will also exhibit a larger variance. As shown in Fig. 1, since the sequential prediction error measures the accumulation of these errors and not their asymptotic value, the fourth-order model indeed has the lowest sequential prediction error of any model order. The PLS criterion selects a model according to its sequential prediction error. This example shows that this is indeed a valid criterion for order estimation.

In this paper, we are mainly interested in prediction (and not in modeling), and we consider the sequential linear prediction problem from a different perspective than is traditionally taken. Rather than focusing on selecting a specific model order and a set of parameters based on their relative performance, we propose a method of prediction based on a weighted combination (or mixture) over all possible predictors. For reasons discussed later, we call this predictor *universal* with respect to both parameters and model orders. We show, basically, that the performance of this predictor is at least as good as the best linear predictor of any model order, even when the parameters are tuned to the data.

The paper begins, in the next section, with a brief background on universal prediction and universal coding. Its purpose is to illustrate several concepts that so far have appeared mainly in the information theory literature. We discuss universal prediction in both a probabilistic setting, where the data is assumed to be an outcome of a stochastic process, and a deterministic setting, where the data is a specific “individual

sequence.” In particular, we discuss the universality of the recursive least squares (RLS) algorithm, which can be used to sequentially achieve the prediction performance of the best fixed-order batch linear predictor, both in the stochastic setting and for every bounded individual sequence. We note that in general, measuring the performance relative to individual sequences is stronger. Our proposed predictor is universal in the stronger setting.

These concepts lay the framework for the main result of the paper, presented in Section III, which is an algorithm for linear prediction that is universal with respect to both the parameters and model order for every individual sequence. The algorithm uses the time-recursive and order-recursive RLS algorithm to generate predictions of the next data point based on the linear models that best fit the data observed so far, of all orders up to some M . Then, it generates a performance-weighted combination of all the various predictors. The universality of this algorithm is shown accordingly. First, as noted above and analyzed in [7], for a given model order, the RLS predictor is universal as it attains sequentially the same accumulated prediction error as a batch predictor. Our main result, then, bounds the additional sequential prediction error incurred by our performance-weighted prediction scheme over the error of the RLS algorithm with the best model order. This excess error, due to the unknown model order, turns out to be negligible.

An important feature of the proposed algorithm is its computational efficiency. As discussed in Section IV, by using an efficient lattice implementation, the proposed universal prediction algorithm has a computational complexity that is not larger than that of the largest model order in the mixture. The development of the universal algorithm does not rely on the problem being one of prediction, and therefore, in Section IV, we also develop a lattice implementation of a universal adaptive equalizer. Examples of the performance of these algorithms are given in Section V, and some concluding remarks are made in Section VI.

II. AN OVERVIEW OF UNIVERSAL PREDICTION

The general universal prediction problem is concerned with the following situation. An observer sequentially receives a sequence of observations $x[1], x[2], \dots, x[t], \dots$. At each time instant t , after having seen $x^{t-1} = x[1], \dots, x[t-1]$ but not $x[t]$, the observer predicts the next outcome $x[t]$ or, more generally, makes a decision b_t based on the observed past x^{t-1} . Associated with this prediction or decision b_t and the actual outcome $x[t]$, there is a loss function $l(b_t, x[t])$ that measures performance. A common example occurs when $b_t = \hat{x}[t]$ is an estimate of $x[t]$ based on x^{t-1} , and $l(b_t, x[t]) = l(\hat{x}[t], x[t])$ is some estimation performance criterion, e.g., the Hamming distance (if $x[t]$ is discrete). In this paper, we consider the squared error $l(b_t, x[t]) = (\hat{x}[t] - x[t])^2$ as the performance criterion.

In the probabilistic setting of the universal prediction problem, it is assumed that the data is governed by some *unknown* probabilistic model P . The objective of a universal predictor is normally to minimize the expected cumulative loss, at least

asymptotically for large n , simultaneously for any source in a certain class. Specifically, a universal predictor $\{b_t^u(x^{t-1})\}$ does not depend on the unknown P , yet it keeps the difference between its average expected loss $E_P\{(1/n)\sum_{t=1}^n l(b_t^u, X_t)\}$ and the optimal average expected loss attained when P is known, vanishingly small for large n .

The simplest situation in the probabilistic setting is universality with respect to an indexed class of sources, where it is assumed that the source is unknown except that it is a member of a certain indexed class $\{P_\theta, \theta \in \Lambda\}$, where Λ is the index set. Most commonly, θ designates a parameter vector of a smooth parametric family, e.g., the families of finite-alphabet memoryless sources, k th-order Markov sources, or $AR(p)$, which is the class of p th-order Gaussian AR sources relevant to this paper. In these parametric cases, we can present universal predictors, and in addition, we can determine, in many cases, the optimal convergence rate to the optimal performance. In many smooth parametric classes and continuous loss functions, this rate behaves as $O(k/2 \cdot \log n/n)$, where k is the number of parameters, and n is the data size.

A more complicated situation is when the source is known to belong to some very large class of sources, e.g., the class of all stationary and ergodic sources. In this class, universality can be shown in many cases, but there is no uniform rate of convergence to the optimal performance. Another class of sources that is relevant for this paper is the class of Markov sources with *unknown* model order k or the class of $AR(p)$ Gaussian sources with *unknown* order p . Again, in these cases, there is no uniform rate of convergence, as the rate can be slow for high-order models. Nonetheless, it turns out that in certain situations, it is possible to achieve a rate that is essentially as small as if k (or p) were known *a priori*. This is achieved by a “twice universal” prediction scheme similar to the scheme suggested later in this paper. The term “twice universal” was coined by Ryabko [8], [9], who also originally suggested such prediction algorithms.

In the deterministic setting of the universal prediction problem, the observed sequence is an individual sequence that is not assumed to be randomly drawn by some probability law. One difficulty associated with this setting is the desired goal. Without any limitations on the class of allowed predictors, there is always the perfect prediction function defined as $b_t(x^{t-1}) = x[t]$, i.e., a predictor tailored to the data. This is a severe overfitting effect to the given data that misses the essence of prediction as a causal, sequential mechanism. Therefore, in this setting, we must limit the class B of allowed predictors in some reasonable way. For example, B could be the class of predictors that are implementable by finite-state machines (FSM’s) with M states or k th-order Markov-structured predictors of the form $b_t(x^{t-1}) = b(x[t-k], \dots, x[t-1])$. There are two relevant classes in this paper; the first is a finite collection of RLS-based predictors, each of a different model order, and the second is the class of fixed predictors of the form $b_t(x^{t-1}) = \sum_{k=1}^p c_k x[t-k]$, i.e., linear predictors of some order p .

In the deterministic setting of universal prediction, the goal is then to perform, for any individual sequence, as well as the best predictor, tuned to that sequence, in some

class. Stated more formally, for a given class B of predictors, we seek a universal sequential predictor $\{b_t^u\}_{t \geq 1}$ whose prediction algorithm is independent of the sequence, yet its average loss $n^{-1} \sum_{t=1}^n l(b_t^u, x[t])$ is asymptotically the same as $\min_B n^{-1} \sum_{t=1}^n l(b_t, x[t])$ for every sequence $x^n = x[1], \dots, x[n]$. The universal predictor need not be necessarily in B , but it must be causal, whereas the reference predictor in B that minimizes the average loss may (by definition) depend on the entire sequence x^n , i.e., be allowed to look at the sequence in advance.

Analogously to the probabilistic case, here we also distinguish between levels of universality, which are now in accordance with the richness of the class B . The first level corresponds to an indexed class of predictors. Examples of this are parametric classes of predictors like finite-state machines with a given number of states, fixed-order Markov predictors, predictors based on neural nets with a given number of neurons, and, of course, fixed-order linear predictors. The rate of convergence depends on the richness of the reference class. A more complex level corresponds to very large classes like the class of all finite-state predictors (without specifying the number of states) operating on infinitely long sequences, etc. Here, uniform rates of convergence may not exist. Finally, as in the probabilistic setting, "twice universal" predictors can be suggested that are universal with respect to a large class of machines, yet their convergence rate, as compared with a more limited class, e.g., linear predictors of order p , depends on the richness of that smaller class. See [10] for a recent survey paper on universal prediction.

We end this section by discussing the recursive least squares (RLS) prediction algorithm as an example of a universal predictor for the class of linear predictors with a given model order p . As noted in the introduction, the RLS algorithm essentially estimates the linear predictor coefficients $c_{p,k}^{t-1}$, $k = 1, \dots, p$ based on all of the data observed up to time $t-1$ by minimizing $\sum_{j=1}^{t-1} (x[j] - \sum_{k=1}^p c_{p,k}^{t-1} x[j-k])^2$. These coefficients are then used to predict the sample $x[t]$ as $\hat{x}_p[t] = \sum_{j=1}^p c_{p,j}^{t-1} x[t-j]$. Once the sample $x[t]$ is observed, the coefficients are then updated to include this sample by minimizing $\sum_{j=1}^t (x[j] - \sum_{k=1}^p c_{p,k}^t x[j-k])^2$. This can be done in a time-recursive efficient way. As defined in (4) above, the resulting sequential prediction error, or "loss," is $l_n(x, \hat{x}_p) = \sum_{t=1}^n (x[t] - \hat{x}_p[t])^2$. The goal of the universal algorithm is to attain the performance of the best algorithm from a certain class, which, in our case, is the class of the p th-order linear predictors. The accumulated error of the best p th-order batch linear predictor is $E_n(x, \hat{x}_p^B) = \sum_{t=1}^n (x[t] - \sum_{j=1}^p c_{p,j}^n x[t-j])^2$ defined in (2). It can be easily shown [7] that for every signal, the sequentially achieved prediction error will be greater than or equal to the batch prediction error, i.e.,

$$l_n(x, \hat{x}_p) \geq E_n(x, \hat{x}_p^B).$$

The interesting result, however, shown in [7], is that for every bounded signal, the RLS algorithm can sequentially achieve the average prediction performance of the batch algorithm to

within an $O(n^{-1} \ln(n))$ term¹

$$\frac{1}{n} l_n(x, \hat{x}_p) \leq \frac{1}{n} E_n(x, \hat{x}_p^B) + O(n^{-1} \ln(n)). \quad (5)$$

Thus, by "plugging in" the best estimate of the predictor coefficients at time $t-1$ to predict $x[t]$, using RLS, we obtain a universal prediction algorithm in the deterministic setting with respect to the class of all fixed linear predictors of order p .

In the stochastic setting, Davisson [11] has shown that for a stationary Gaussian time series, the expected squared sequential prediction error for a linear predictor of order p given data up to time n is

$$\sigma^2[p; t] = E\{x[t] - \hat{x}_p[t]\}^2 = \sigma^2[p; \infty] \left(1 + \frac{p}{t}\right) + o(t^{-1}) \quad (6)$$

where $\sigma^2[p; \infty] = \lim_{t \rightarrow \infty} \sigma^2[p; t]$, which exists and is the optimal expected square error without the sequentiality constraint, i.e., the batch error. Thus, by calculating the harmonic sum of terms of the form p/t , the time-averaged accumulation of the additional prediction error of an RLS type algorithm over the batch error is approximately $p \ln n/n$ for data of size n . This establishes the universality and the convergence rate of the prediction algorithm based on RLS in the stochastic Gaussian setting.

III. MAIN RESULTS

The main contribution of this paper is a twice universal linear prediction algorithm that does not fix the order in advance, but rather weights all possible model orders according to their performance so far. The accumulated average square error $(1/n)l_n(x, \hat{x})$ of this algorithm is better, to within a negligible term, than that of an RLS predictor whose order was preset to p for any p less than some $M < \infty$. Since the RLS algorithm of order p outperforms any fixed linear predictor of order p , our algorithm attains asymptotically the performance of the best fixed (or sequential) linear predictor of any order less than M . In our derivation, we only assume that the predicted sequence $x[1], x[2], \dots$ is bounded, i.e., $|x[t]| < A < \infty$ for all t , but is otherwise an arbitrary, real-valued sequence.

An explicit description of the universal predictor we suggest is as follows. Let $\hat{x}_k[t]$ be the output of a sequential linear predictor of order k , as obtained by the RLS algorithm with model order k . The universal prediction at time t , $\hat{x}_u[t]$, is a performance-weighted combination of the outputs of each of the different sequential linear predictors of orders 1 through M , i.e.,

$$\hat{x}_u[t] = \sum_{k=1}^M \mu_k[t] \hat{x}_k[t]$$

where

$$\mu_k[t] = \frac{\exp\left(-\frac{1}{2c} l_{t-1}(x, \hat{x}_k)\right)}{\sum_{j=1}^M \exp\left(-\frac{1}{2c} l_{t-1}(x, \hat{x}_j)\right)}$$

¹The impact of the model order was not accurately determined in [7]; however, a careful straightforward calculation can lead to an $O(n^{-1} p^3 \ln(n))$ excess loss term.

and c is a constant parameter to be defined later. Our main theorem (and its corollary) below relates the performance of the universal predictor

$$l_n(x, \hat{x}_u) = \sum_{t=1}^n (x[t] - \hat{x}_u[t])^2$$

to that of the best sequential and batch predictors of order less than M .

Theorem 1: Let $x[n]$ be a bounded real-valued arbitrary sequence such that $|x[n]| < A$. Then, $l_n(x, \hat{x}_u)$ satisfies

$$\frac{1}{n} l_n(x, \hat{x}_u) \leq \min_k \frac{1}{n} l_n(x, \hat{x}_k) + \frac{8A^2}{n} \ln(M).$$

Corollary 1:

$$\begin{aligned} & \frac{1}{n} l_n(x, \hat{x}_u) \\ & \leq \min_k \left\{ \frac{1}{n} E_n(x, \hat{x}_k^B) + \frac{8A^2}{n} \ln(M) + O\left(\frac{\ln(n)}{n}\right) \right\}. \end{aligned}$$

The corollary follows from the theorem and from (5).

The theorem and its corollary tell us that the average squared prediction error of the universal prediction algorithm is within $O(n^{-1})$ of the best sequential linear prediction algorithm and within $O(n^{-1} \ln(n))$ of the best batch linear prediction algorithm uniformly for every individual sequence $x[n]$. As we will see, the cost terms can be identified as a model redundancy term proportional to $n^{-1} \ln(M)$ due to the lack of knowledge of the best model order, plus a parameter redundancy term proportional to $n^{-1} \ln(n)$ due to the lack of knowledge of the parameters and the learning time of RLS.

Regarding the parameter redundancy term, which is a result of applying the RLS algorithm to individual sequences, we have noted that following the analysis in [7], its dependence on the model order k is of the form $O(k^3 \ln(n)/n)$. However, in the stochastic case, as implied both by Davissou's result and Rissanen's lower bound given in the general MDL context [12], this redundancy is only $O(k \ln(n)/n)$. If the bound derived by the technique of [7] is tight, it suggests that the approach to "plug-in" the previous best parameters to predict the next data point used by RLS is probably not the best thing to do. In a current work (see [13]), we suggest a double mixture approach over model orders and parameters to achieve an $O(k \ln(n)/n)$ bound. This may be indicative of a new direction for adaptive parameter estimation based on mixture parameter models.

Returning to the theorem, the basic idea behind its proof is the following. We define a "probability" assignment of each predictor to the data sequence x^n such that the probability is an exponentially decreasing function of the total squared error for that predictor. This use of prediction error as a probability or likelihood was also suggested by Rissanen [12] and Vovk [14]. By defining a universal probability as an *a priori* average of the assigned probabilities, then to first order in the exponent, the universal probability will be dominated by the largest exponential, i.e., the probability assignment of the model order with the smallest total squared error. By relating back the universal probability assignment to the accumulated

squared error of the universal predictor, we get the desired result.

Proof of the Theorem: Suppose a set of sequential linear predictors of order k , $1 \leq k \leq M$ are given, and the loss of each is $l_n(x, \hat{x}_k)$ defined in (4). We define the following function of the loss of the k th-order predictor

$$P_k(x^n) \triangleq B \exp\left(-\frac{1}{2c} l_n(x, \hat{x}_k)\right)$$

which can be viewed as a probability assignment of the k th-order model to the data $x[t]$ for $0 \leq t \leq n$. We also define a conditional probability

$$P_k(x[n] | x^{n-1}) = \frac{P_k(x^n)}{P_k(x^{n-1})} = \exp\left(-\frac{1}{2c} l(x[n], \hat{x}_k[n])\right)$$

where the notation $l(x[n], \hat{x}_k[n])$ is taken to mean the instantaneous loss at time n , i.e., $(x[n] - \hat{x}_k[n])^2$. Hence, the probability assigned to the entire data sequence is simply a product of the conditional probabilities. Define the universal probability $P_u(x^n)$ as

$$P_u(x^n) = \sum_{i=1}^M w_i P_i(x^n)$$

where $\sum_i w_i = 1$. For this proof, we use uniform *a priori* weights $w_i = 1/M$; however, the proof can be constructed with other weights leading to a slightly different "redundancy term." This $P_u(x^n)$ yields a conditional probability

$$\begin{aligned} P_u(x[n] | x^{n-1}) &= \frac{\frac{1}{M} \sum_{i=1}^M P_i(x^n)}{\frac{1}{M} \sum_{j=1}^M P_j(x^{n-1})} \\ &= \frac{\sum_{i=1}^M P_i(x[n] | x^{n-1}) P_i(x^{n-1})}{\sum_{j=1}^M P_j(x^{n-1})} \\ &= \sum_{i=1}^M \mu_i(n) P_i(x[n] | x^{n-1}) \end{aligned}$$

where

$$\mu_i[n] = \frac{P_i(x^{n-1})}{\sum_{j=1}^M P_j(x^{n-1})}.$$

Note that the conditional universal probability $P_u(x[n] | x^{n-1})$ is a weighted average of the M conditional probabilities $P_i(x[n] | x^{n-1})$, where the weights $\mu_i[n]$ are proportional to $P_i(x^{n-1})$: the performance of the i th model on the data through time $n - 1$.

By the definition of $P_u(x^n)$, we have

$$P_u(x^n) \geq \max_i \frac{1}{M} P_i(x^n)$$

which leads to

$$\begin{aligned}
 -\ln(P_u(x^n)) &\leq \min_i \{\ln(M) - \ln(P_i(x^n))\} \\
 &\leq \min_i \left\{ \ln(M) - \ln B + \frac{1}{2c} l_n(x, \hat{x}_i) \right\} \quad (7)
 \end{aligned}$$

relating the negative of the logarithm of the universal probability to the total squared error of the best linear predictor, i.e., the minimum loss over i . However, how is this related to $l_n(x, \hat{x}_u)$, which is the total squared error of the universal predictor? To answer this, we define another ‘‘probability’’ in terms of the performance of the universal predictor

$$\begin{aligned}
 \tilde{P}_u(x^n) &\triangleq B \exp\left(-\frac{1}{2c} l_n(x, \hat{x}_u)\right) \\
 &= B \exp\left(-\frac{1}{2c} \sum_{t=1}^n \left(x[t] - \sum_{i=1}^M \mu_i[t] \hat{x}_i[t]\right)^2\right) \\
 &= \prod_{t=1}^n B \exp\left(-\frac{1}{2c} \left(x[t] - \sum_{i=1}^M \mu_i[t] \hat{x}_i[t]\right)^2\right) \\
 &= \prod_{t=1}^n f_t\left(\sum_{i=1}^M \mu_i[t] \hat{x}_i[t]\right) \quad (8)
 \end{aligned}$$

where $f_t(\cdot)$ is defined as

$$f_t(z) \triangleq B \exp(-(x[t] - z)^2/2c). \quad (9)$$

However

$$P_u(x^n) = \prod_{t=1}^n \sum_{i=1}^M \mu_i[t] f_t(\hat{x}_i[t]) \quad (10)$$

where $\sum_{i=1}^M \mu_i[t] = 1$. Note that in (8), $\tilde{P}_u(x^n)$ is a product of a function evaluated at a convex combination of values, whereas in (10), $P_u(x^n)$ is a product of a convex combination of the same function evaluated at the same values. If the function $f_t(\cdot)$ is concave and $\sum_i \theta_i = 1$, then

$$f_t\left(\sum_{i=1}^M \theta_i z_i\right) \geq \sum_{i=1}^M \theta_i f_t(z_i)$$

by Jensen’s inequality. The function $f_t(\cdot)$ as defined in (9) will be concave for any values z_i such that $z_i^2 < c$. This corresponds to

$$-\sqrt{c} \leq (x[t] - \hat{x}_i[t]) \leq \sqrt{c}. \quad (11)$$

Since the signal $|x[t]| < A$, then the linearly predicted values $\hat{x}_i[t]$ can always be chosen such that $|\hat{x}_i[t]| < A$. If the linearly predicted values are outside this range, then the prediction error can only decrease by clipping. Therefore, by Jensen’s inequality, whenever

$$c \geq 4A^2$$

the function $f(\cdot)$ will be concave at all of the points $\hat{x}_i[t]$ and

$$\tilde{P}_u(x^n) \geq P_u(x^n). \quad (12)$$

Whenever (12) holds, it yields, with (7)

$$-\ln \tilde{P}_u(x^n) \leq -\ln P_u(x^n) \leq \min_i \frac{1}{2c} l_n(x, \hat{x}_i) + \ln(M) - \ln B.$$

Since $-\ln \tilde{P}_u(x^n) = (1/2c)l_n(x, \hat{x}_u) - \ln B$

$$\frac{1}{2c} l_n(x, \hat{x}_u) \leq \min_i \frac{1}{2c} l_n(x, \hat{x}_i) + \ln(M)$$

or

$$\frac{1}{n} l_n(x, \hat{x}_u) \leq \min_i \frac{1}{n} l_n(x, \hat{x}_i) + \frac{2c}{n} \ln(M).$$

The proof is completed by choosing $c = 4A^2$, which is the smallest value that guarantees, without further assumptions, that the concavity condition holds. ■

As noted in the proof, the model order redundancy term $8A^2 \ln(M)/n$ can be improved upon. Rather than using *a priori* weights $w_i = 1/M$, we could have weighted each of the models inversely proportional to their model order, i.e.,

$$w_i = \frac{i^{-1}}{\sum_{j=1}^M j^{-1}} > \frac{1}{i(\ln(M) + 1)}.$$

The proof remains intact with the model order redundancy term being $-\ln(w_p)/n$ rather than $-\ln(1/M)/n$, where p is the order of the model with the smallest prediction error. The resulting model order redundancy term becomes $8A^2(\ln(p)/n) + \ln(\ln(M) + 1)/n$. We can also relax the assumption that there is a finite known largest order M by using an *a priori* weight distribution $\{w_i\}$ defined over all the integers (e.g., the universal probability over the integers suggested in [15]), although such a choice requires a computationally more complex algorithm.

An important factor that affects both the algorithm and the convergence rate is the choice of c . Condition (11) requires only that c upper bounds the square error of the largest prediction error. We have taken a ‘‘worst case’’ cautious approach and chose $c = 4A^2$; however, in many cases, we can assume that the maximal prediction error is less than $2A$, and therefore, c can be smaller, leading to a smaller ‘‘redundancy term.’’

Finally, we note that the technique presented here actually shows the following more general result. Suppose we have a set of arbitrary M predictors. The accumulated square error of a predictor that uses a performance-weighted combination of the output of these predictors is larger by at most $8A^2 \log M$ than the best predictor in this set in predicting any bounded sequence. In other words, we have shown a universal predictor that outperforms a set of ‘‘experts’’ (see [16] for the problem definition). Universal prediction in this setting, with the square error loss, was also discussed in [17]. The resulting extra loss of the universal predictor suggested there is even better by a factor of four than our predictor. However, the proposed algorithm, based on the Vovk procedure [14], is more complicated and cannot be represented as a weighted combination of the ‘‘experts’’ predictions.

TABLE I

UNIVERSAL LINEAR PREDICTION ALGORITHM BASED ON THE LEAST-SQUARES LATTICE ALGORITHM FOR TIME- AND ORDER-RECURSIVE COMPUTATION OF THE PREDICTOR OUTPUTS. THE INPUT SIGNAL $x[n]$ IS ASSUMED BOUNDED SUCH THAT $|x[n]| < A$ FOR ALL n . THE AVERAGE SQUARED PREDICTION ERROR OF THE OUTPUT $\hat{x}_u[n]$ IS WITHIN $O(A^2 \ln(M)/n)$ OF THE BEST MODEL ORDER LESS THAN M UNIFORMLY FOR EVERY SIGNAL

-
- Initialize:
 - $\rightarrow \gamma_m[-1] = r_m[-1] = K_{m+1}[-1] = 0$, for $0 \leq m < M$
 - For each time $n \geq 0$ compute:
 - $\rightarrow \gamma_0[n] = 0$
 - $\rightarrow e_0[n] = r_0[n] = x[n]$
 - $\rightarrow \epsilon_0^e[n] = \epsilon_0^r[n] = w\epsilon_0^e[n-1] + x^2[n]$
 - $\rightarrow \hat{x}_0[n+1] = 0$
 - For each model order, $0 \leq m < M$ compute:
 - $\rightarrow K_{m+1}[n] = wK_{m+1}[n-1] + e_m[n]r_m[n-1]/(1-\gamma_m[n-1])$
 - $\rightarrow k_{m+1}^e[n] = K_{m+1}[n]/\epsilon_m^e[n]$, $k_{m+1}^r[n] = K_{m+1}[n]/\epsilon_m^r[n-1]$
 - $\rightarrow e_{m+1}[n] = e_m[n] - k_{m+1}^r[n]r_m[n-1]$
 - $\rightarrow r_{m+1}[n] = r_m[n-1] - k_{m+1}^e[n]e_m[n]$
 - $\rightarrow \epsilon_{m+1}^e[n] = \epsilon_m^e[n] - k_{m+1}^r[n]K_{m+1}[n]$
 - $\rightarrow \epsilon_{m+1}^r[n] = \epsilon_m^r[n-1] - k_{m+1}^e[n]K_{m+1}[n]$
 - $\rightarrow \gamma_{m+1}[n] = \gamma_m[n] + \frac{r_m[n]^2}{\epsilon_m^e[n]}$
 - For each model order, $0 \leq m < M$ compute:
 - $\rightarrow \hat{x}_{m+1}[n+1] = \hat{x}_m[n+1] + k_{m+1}^r[n]r_m[n]/(1-\gamma_m[n])$
 - $\rightarrow l_n(x, \hat{x}_{m+1}) = l_{n-1}(x, \hat{x}_{m+1}) + (x[n] - \hat{x}_{m+1}[n])^2$
 - $\rightarrow \mu_{m+1}[n+1] = \exp(-l_n(x, \hat{x}_{m+1})/2c) / \sum_{k=1}^M \exp(-l_n(x, \hat{x}_k)/2c)$, $c = 4A^2$
 - Compute the universal predictor output:
 - $\rightarrow \hat{x}_u[n+1] = \sum_{m=1}^M \mu_m[n+1]\hat{x}_m[n+1]$
-

IV. ALGORITHMIC ISSUES

The main result of this paper, as stated in Theorem 1, bounds the prediction performance of the universal predictor to within a model order redundancy term and a parameter redundancy term from the performance of the best batch algorithm for linear prediction. An issue that remains is the computational complexity of the universal approach, which requires the predicted values from each of the model orders and their sequential prediction error to compute each predicted value. At first glance, it might appear that the computational cost of our universal predictor is rather high, requiring the solution of each of the linear prediction problems $i = 1, \dots, M$ in parallel. However, the linear prediction problems for each model order have a great deal in common with one another, and this structure can be exploited. Indeed, just as the RLS algorithm for a given model order can be written as a time recursion, there exist time- and order-recursive solutions to the least-squares prediction problem in which at each time step, the M th-order prediction problem can be constructed by recursively solving for each of the predictors of lower order. The resulting complexity of these algorithms can be made to have $O(M)$ operations per time sample, which results in a total complexity of $O(Mn)$. See, for example, the least-squares lattice algorithms in [18]–[23]. Although the universal

predictor can be computed using any one of a large class of RLS algorithms, for completeness, one such least-squares lattice algorithm from [24] is presented in Table I, along with the modifications necessary to compute the universal predictor output. This algorithm is based on a prewindowed least-squares lattice algorithm with *a posteriori* residuals. In order to compute the *a priori* predictions of each of the different model orders and the universal predictor output, the last four equations have been added. A forgetting factor $w \leq 1$ has been included to emphasize the most recent data in the calculation of the parameters. Setting $w = 1$ corresponds to the growing memory least-squares prediction problem. To compute the exact least-squares solution, successive stages of the lattice must be “turned on” at each time for $n < M$, i.e., the order recursions are computed up to order n for $n < M$. An alternative initialization, which is often used, is to set the cost functions $\epsilon_n^e[0]$ and $\epsilon_m^r[0]$ to a small constant $\delta > 0$ to ensure that the algorithm is stable, and then, the order recursions can be computed for all m at each time. This does not produce an exact least-squares solution; however, it is generally very close to the exact solution.

This algorithm can be viewed as operating M separate adaptive filters, or linear predictors, and combining their results with a performance-weighted average. At each time,

TABLE II

UNIVERSAL EQUALIZATION ALGORITHM BASED ON THE LEAST-SQUARES ADAPTIVE LATTICE EQUALIZATION ALGORITHM FOR TIME- AND ORDER-RECURSIVE COMPUTATION OF THE LATTICE EQUALIZER PARAMETERS. THE INPUT SIGNAL $x[n]$ IS ASSUMED BOUNDED SUCH THAT $|x[n]| < A$ FOR ALL n . THE AVERAGE SQUARED EQUALIZATION ERROR AFTER TRAINING OF THE OUTPUT $\hat{y}_u[n]$ IS WITHIN $O(A^2 \ln(M)/n)$ OF THE BEST MODEL ORDER LESS THAN M UNIFORMLY FOR EVERY SIGNAL

For each time $n \geq 0$ compute:

• Initialize:

$$\begin{aligned} \rightarrow e_0[n] &= r_0[n] = x[n] \\ \rightarrow \epsilon_0^e[n] &= \epsilon_0^r[n] = w\epsilon_0^e[n-1] + x^2[n] \\ \rightarrow y_{-1}[n] &= \gamma_{-1}[n] = \gamma_{-1}[n-1] = 0 \\ \rightarrow \bar{e}_{-1}[n] &= a[n] \end{aligned}$$

• For each model order, $m = 0, \dots, M-1$ compute:

$$\begin{aligned} \rightarrow k_m[n] &= wk_m[n-1] + (1 - \gamma_{m-1}[n-1])e_m[n]r_m[n-1] \\ \rightarrow e_{m+1}[n] &= e_m[n] - k_m[n-1]r_m[n-1]/\epsilon_m^r[n-2] \\ \rightarrow r_{m+1}[n] &= r_m[n-1] - k_m[n-1]e_m[n]/\epsilon_m^e[n-1] \\ \rightarrow \epsilon_{m+1}^e[n] &= \epsilon_m^e[n] - k_m^2[n]/\epsilon_m^r[n-1] \\ \rightarrow \epsilon_{m+1}^r[n] &= \epsilon_m^r[n-1] - k_m^2[n]/\epsilon_m^e[n] \\ \rightarrow \gamma_m[n] &= \gamma_{m-1}[n] + ((1 - \gamma_{m-1}[n])r_m[n])^2/\epsilon_m^r[n] \end{aligned}$$

• For each model order, $m = 0, \dots, M$ compute:

$$\begin{aligned} \rightarrow y_m[n] &= y_{m-1}[n] + \bar{k}_m[n-1]r_m[n]/\epsilon_m^r[n-1] \\ \rightarrow \bar{e}_m[n] &= a[n] - y_m[n] \\ \rightarrow l_n(a, y_m) &= l_{n-1}(a, y_m) + \bar{e}_m^2[n] \\ \rightarrow \bar{k}_m[n] &= w\bar{k}_m[n-1] + (1 - \gamma_{m-1}[n])\bar{e}_{m-1}[n]r_m[n] \\ \rightarrow \mu_m[n] &= \exp(-l_{n-1}(a, y_m)/2c) / \sum_{k=1}^M \exp(-l_{n-1}(a, y_k)/2c), \quad c = 4A^2 \end{aligned}$$

• Finally

$$\rightarrow \hat{y}_u[n] = \sum_{m=1}^M \mu_m[n]y_m[n]$$

the universal predictor weights each of the separate predicted values by $\mu_m[\ell]$, which is proportional to $\exp(-l_{t-1}(x, \hat{x}_m))$. As a result, each of the different model orders compete for a contribution to the output, with their contributions depending exponentially on their cumulative sequential performance. If any of the model orders outperforms the others, then its weight will be exponentially larger than the rest. However, the model order with the best cumulative performance can change over the length of the data, giving more weight to models of different orders with time.

The inclusion of an adaptation parameter or forgetting factor can be used to accommodate slowly time-varying signals. Here, the parameters of the predictor for each model order are calculated with an exponentially decreasing emphasis on the past. As a result, the parameters reflect the most recent data over an "effective window" of length $1/(1-w)$. If the mixture weights of the universal predictor are computed using the accumulated square-error, i.e., $l_n(x, \hat{x}_k)$, then regardless of how the parameters are selected for each model order, by Theorem 1, the accumulated square error for the universal predictor will be within $O(\ln(M)/n)$ of the performance of the best model order. However, if the mixture weights are computed using the adaptation parameter

$$l_n^w(x, \hat{x}_m) = wl_{n-1}(x, \hat{x}_m) + (x[n] - \hat{x}_m[n])^2$$

then the results of Theorem 1 still hold with the performance measured by $l_n^w(x, \hat{x}_u)$, that is

$$\frac{1}{n}l_n^w(x, \hat{x}_u) \leq \min_k \frac{1}{n}l_n^w(x, \hat{x}_k) + \frac{8A^2}{n} \ln(M).$$

As is often used in adaptive filtering applications, a finite-length sliding window, such as a Hamming window [20], can be applied to the data and the performance measure with the results of Theorem 1 remaining intact.

There is nothing in the development of Theorem 1 that requires that the outputs of the adaptive filters be predictions of the input signal. All that is required is that the performance metric among several candidate algorithms is one of sequentially accumulated squared error. The main result is actually more general in that it applies to any sequential decision problem in which several candidate approaches are compared using their sequentially accumulated squared errors. As an example of another application of this result, an adaptive equalization algorithm can be developed as a direct analog of the prediction algorithm. For example, suppose that a data sequence $a[n]$ is transmitted over a noisy channel such that the received signal $x[n]$ could be represented as

$$x[n] = \sum_{k=1}^P h[k]a[n-k] + w[n]$$

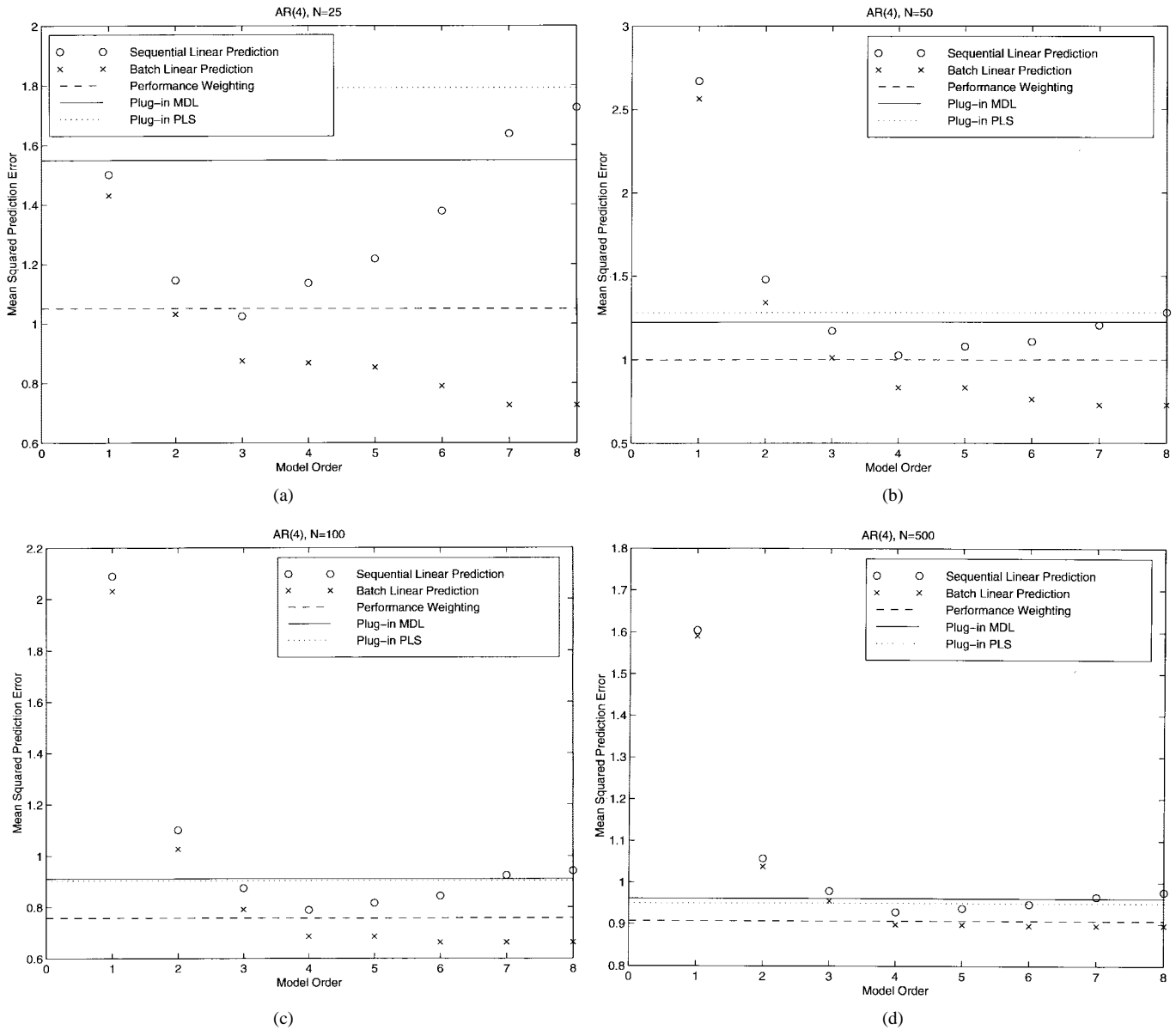


Fig. 2. Prediction results for a sample function of the fourth-order AR process (13). The average squared sequential prediction error $l_n(x, \hat{x}_p)$ and the associated batch prediction errors $E_p[n]$ for each of the p th-order linear predictors $p = 1, \dots, 8$ are indicated with “o” and “x” marks, respectively. The prediction errors resulting from “plug-in” of the MDL-order predictor and the PLS-order predictor at each time step are indicated by the solid and dotted lines, respectively. The prediction performance of the universal predictor with performance weighting is indicated by the dashed line. (a) Twenty-five samples. (b) Fifty samples. (c) One hundred samples. (d) Five hundred samples.

where the impulse response of the channel $h[n]$ represents a convolutional distortion, and the signal $w[n]$ corresponds to additive noise. Consider a loss function

$$l_n(a, y_m) = \sum_{t=1}^n (a[t] - y_m[t])^2$$

where $x[n]$ is the input, and $y_m[n]$ is the output of the m th-order least-squares equalizer for data $a[n]$ corresponding to the output total squared equalization error for an equalizer of order m . An algorithm that generates a performance-weighted average of the outputs of all linear equalizers of order less than M can be constructed by similar means to the universal predictor. Since lattice methods also exist for a variety of adaptive filtering applications, including equalization, the

outputs of each of the equalizers of order less than some M can all be constructed recursively. The computational cost of the algorithm is once again only as large as that for the largest model order, i.e., $O(M)$. For simplicity, we only consider real-valued scalar data, although generalization of the lattice methods to complex vector data is straightforward, as would be required to implement decision-feedback or use multichannel data [22], [25]. As an example, we modify the least-squares adaptive lattice equalizer of [26] to construct a universal adaptive equalizer in Table II. The algorithm takes as input a received signal $x[n]$, a training data sequence $a[n]$, and a maximum model order M . A tracking parameter $w \leq 1$ has been included to track small variations in the channel impulse response. Setting $w = 1$ corresponds to the growing memory least-squares equalization problem. When the equalizer is

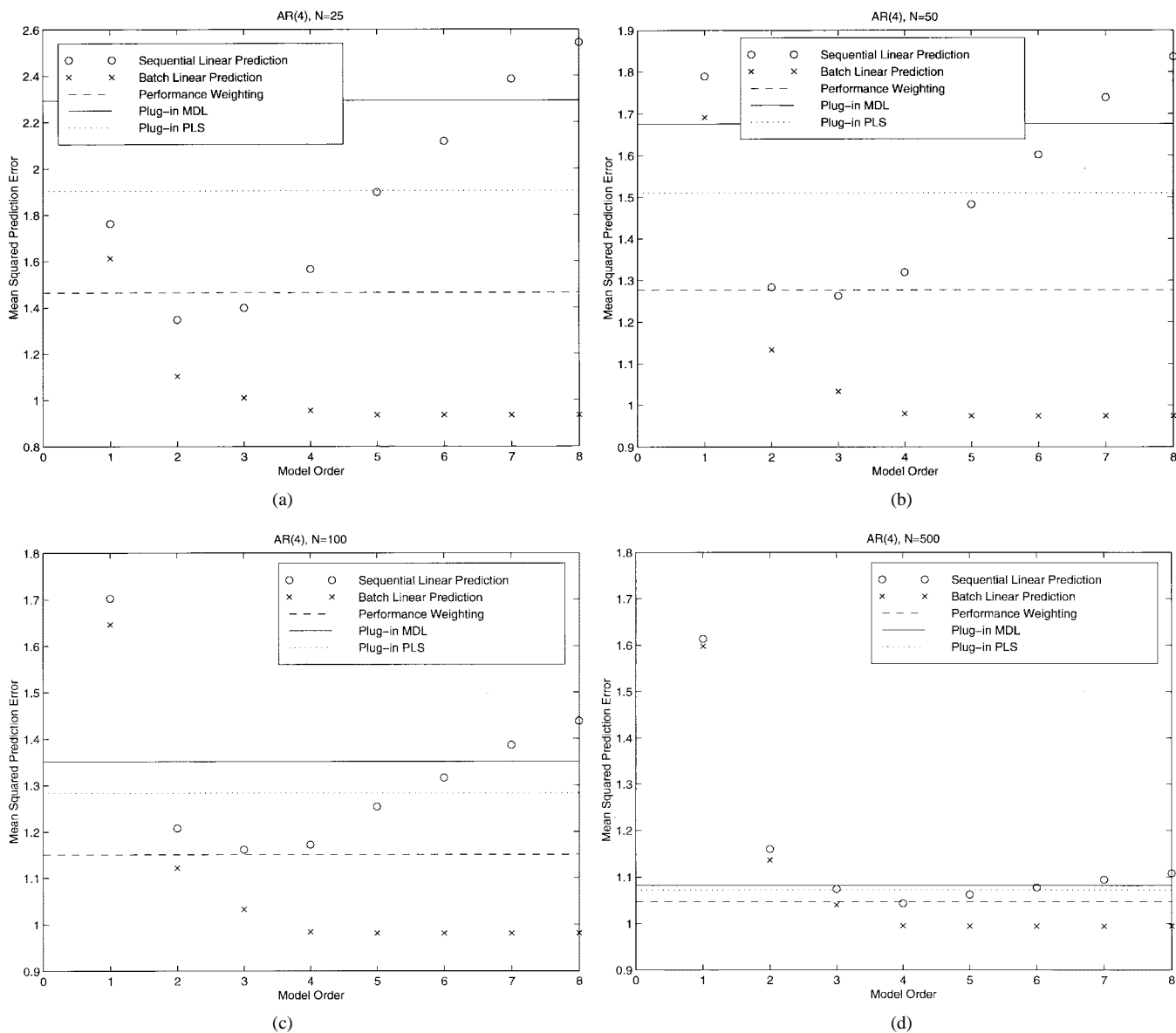


Fig. 3. Average prediction results for 100 different sample functions of the fourth-order AR process (13). The average squared sequential prediction error $I_n(x, \hat{x}_p)$ and the associated batch prediction errors $E_p[n]$ for each of the p -th-order linear predictors $p = 1, \dots, 8$ are indicated with “o” and “x” marks, respectively. The prediction errors resulting from “plug-in” of the MDL-order predictor and the PLS-order predictor at each time step are indicated by the solid and dotted lines, respectively. The prediction performance of the universal predictor with performance weighting is indicated by the dashed line. (a) Twenty-five samples. (b) Fifty samples. (c) One hundred samples. (d) Five hundred samples.

operating on a training sequence $a[n]$, Table II provides the proper update formulas. To run in decision-directed mode on transmitted data, $a[n]$ could be replaced by a suitably quantized $\hat{a}[n] = Q(\hat{y}_u[n])$ or $Q(y_k[n])$.

V. EXAMPLES

We illustrate the performance of the universal linear prediction algorithms developed in this paper with several examples of signal prediction and data equalization. The first set of examples involve the prediction of sample functions from the fourth-order autoregressive process described by

$$x[n] = 0.9x[n - 1] - 0.25x[n - 2] - 0.1x[n - 3] - 0.2x[n - 4] + w[n] \quad (13)$$

where $w[n]$ is a sample function from a stationary white Gaussian noise process with unit variance. Since the main result of this paper governs the performance of the prediction algorithm for any particular individual signal, Fig. 2 shows the running average squared prediction error for a single sample function from (13). The performance-weighted universal prediction algorithm developed in Section IV and given in Table I was used for a single realization of $x[n]$. The parameter A was set to 4; however, the performance is relatively insensitive to changes in A . Although the process is actually of fourth order, as indicated in Fig. 2(a), initially ($N = 25$), the third order sequential linear predictor outperforms each of the other sequential predictors. As the data length is increased, the fourth-order predictor begins to outperform the others. For data lengths of 50, 100, and 500 samples, the universal

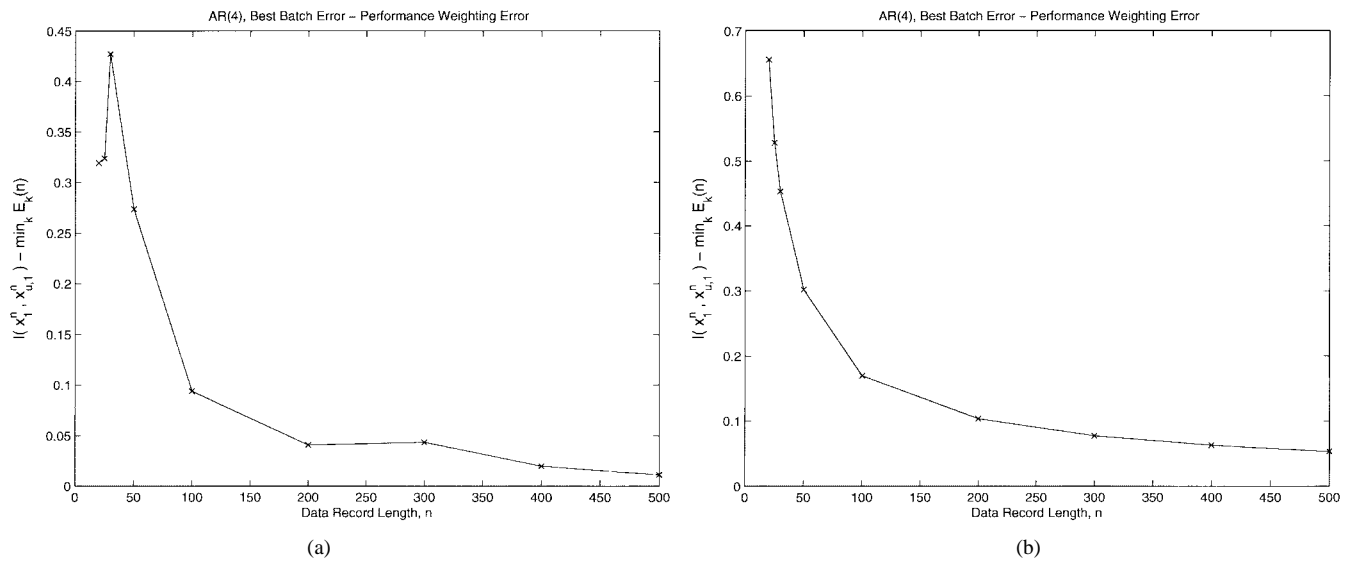


Fig. 4. Difference between the average prediction error of the universal predictor and the best batch predictor for a single sample function and for an average over an ensemble of 100 sample functions of (13) are shown as a function of the length of the data record. The “x” marks indicate the data points of Figs. 2 and 3; the lines are added as visual aids only. (a) Individual sequence. (b) Ensemble average.

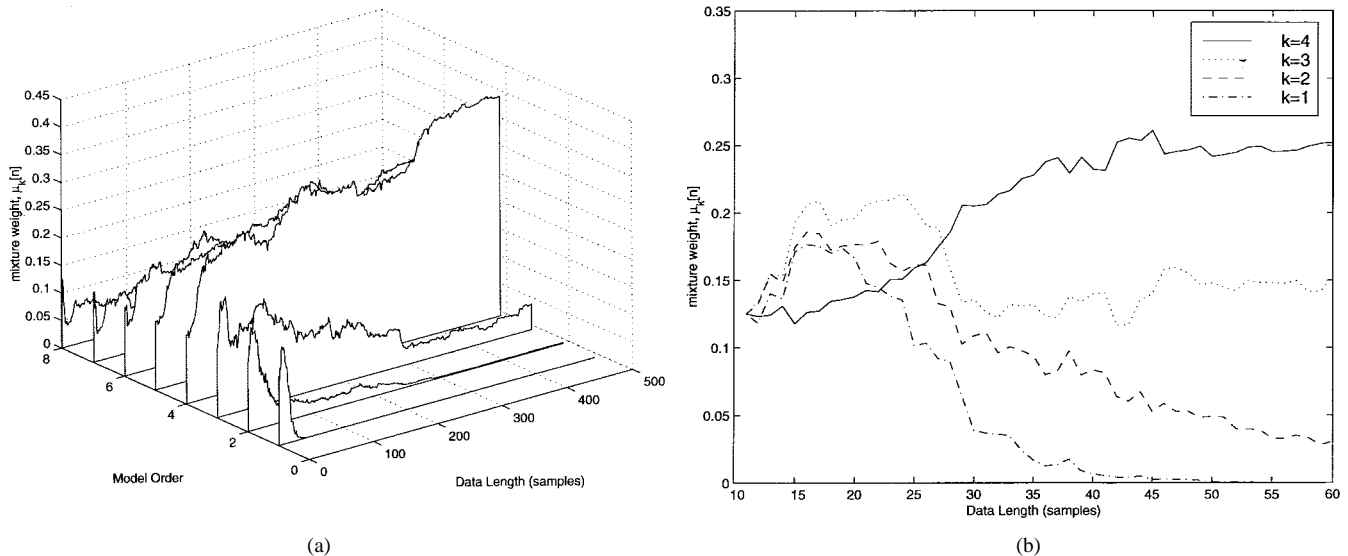


Fig. 5. Mixture weights $\mu_k[n]$ in the universal predictor of Table I are shown as a function of time and model order for a fourth-order autoregressive process. (a) Mixture weights versus time and model order. (b) Model orders 1 through 4 versus time.

algorithm, which is depicted by the dashed line, outperforms all of the model orders, and for $N = 25$, the performance is very close to the best model order. In the figure, the performance of “plug-in” approaches using the MDL and PLS criteria are also shown. At each time sample, the model order indicated by the corresponding order-estimate was used to predict the current sample. For brevity, we refer to the MDL estimate as the model order with the minimum batch prediction error plus linear penalty term and the PLS estimate as the minimum sequential prediction error. The performance-weighted universal approach appears particularly useful for short data records or during the startup or learning time of the individual sequential predictors. Note that the final prediction error of this individual sequence appears to be around 0.9 rather than 1, as might be expected from (13). Regardless of

the value of the minimum error and of which model order achieves it, this universal algorithm is able to adaptively select among the best-performing candidate algorithms. This makes it attractive for adaptive processing in time-varying environments for which a windowed version of the most recent data is typically used [20]. Such applications require that algorithms continually operate in the short effective data-length regime.

In Fig. 3, similar results to those in Fig. 2 are presented and averaged over 100 different sample functions from (13). The ensemble average performance and rates for the autoregressive process are characteristically similar to those for a given sample function. However, for shorter data records, the plug-in approaches appear to be considerably worse on average than indicated in Fig. 2. In addition, the sequential algorithms

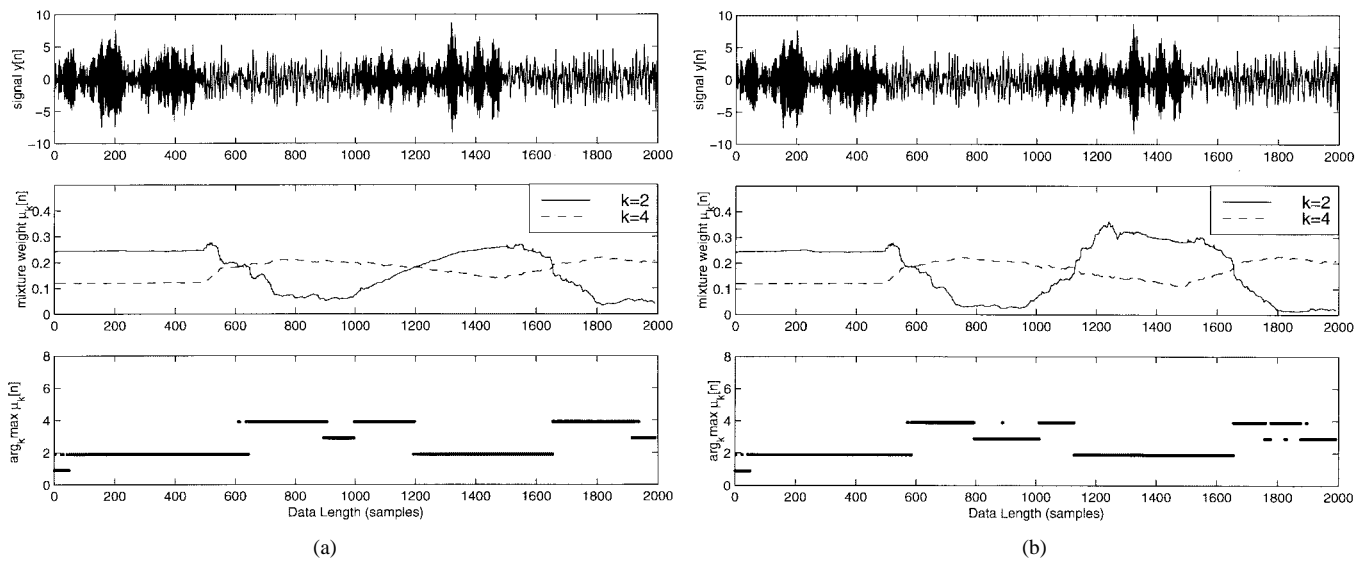


Fig. 6. Top figure shows an autoregressive process that switches between a second- and fourth-order process every 500 samples. The middle figure shows the mixture weights $\mu_k[n]$, $k = 2, 4$ in the universal predictor of Table I for the exponential window and sliding rectangular window. The bottom figure shows the index of model order that has the largest weight as a function of time. (a) Exponential window. (b) Sliding rectangular window.

exhibit a more distinct minimum running average prediction error as a function of model order.

In Figs. 2 and 3, the universal algorithm is shown to achieve the performance of the best of the sequential linear prediction algorithms. As the data record increases, the universal algorithm also attains the performance of the best “batch” algorithm. Although the sequential linear predictors will also asymptotically achieve their corresponding batch performance, the rate at which the universal algorithm achieves the best batch performance is at least as fast by Theorem 1 and its corollary. In Fig. 4, the rate at which the universal algorithm approaches the best batch performance is shown as a function of the data length. By Theorem 1 and its corollary, this rate is at most $O(\ln(n)/n)$.

To further illustrate the operation of the model order mixture in the universal predictor, Fig. 5 depicts the mixture weights $\mu_k[n]$ as a function of time and model order during the prediction of the fourth-order autoregressive process used to generate Fig. 2. The waterfall plot in Fig. 5(a) depicts the progression of each of the mixture weights and illustrates how the weights initially favor lower model orders until the fourth-order model eventually outperforms and outweighs the rest. Fig. 5(b) focuses on the first 50 samples of operation and demonstrates how initially, the second- and third-order models receive the largest weight for the first few samples. The third-order model dominates from about the fifth through the 17th sample, after which the fourth-order model receives the largest weight. Note that for stability purposes, the operation of the universal predictor was started after the tenth sample.

The algorithm described in Table I corresponds to a growing memory implementation of the RLS algorithm. This means that the number of data samples used to compute the prediction parameters increases as a function of time. To accommodate time-varying signals, such growing memory algorithms typically use a tracking parameter $w < 1$, as indicated in Table I. This enables the parameters of the predictor to track

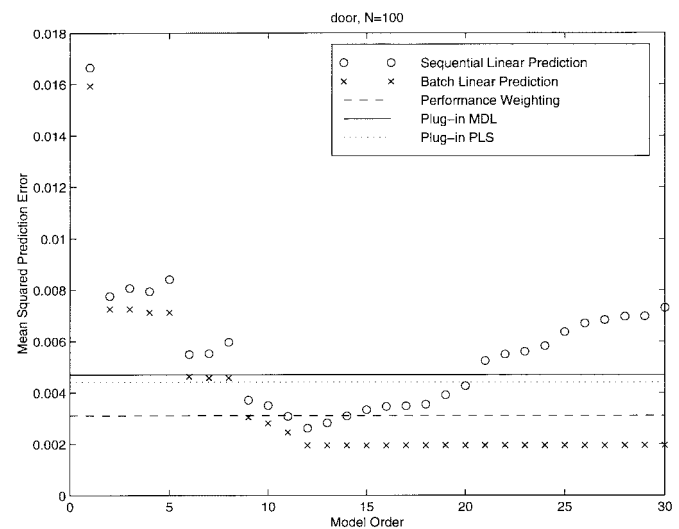


Fig. 7. Prediction results for a 10 ms segment of speech for the word “door” recorded at 10 kHz or 100 samples. The average squared sequential prediction error $l_n(x, \hat{x}_p)$ and the associated batch prediction errors $E_p[n]$ for each of the p th-order linear predictors $p = 1, \dots, 30$ are indicated with “o” and “x” marks, respectively. The prediction errors resulting from “plug-in” of the MDL-order predictor and the PLS-order predictor at each time step are indicated by the solid and dotted lines, respectively. The prediction performance of the universal predictor with performance-weighting is indicated by the dashed line.

slow variations in the process by emphasizing the most recent samples in the data history. If a tracking parameter is used, the weighting that is applied to the data corresponds to an exponential window, where the distant past, say, a sample at $n = n_0$, is weighted exponentially as a function of its distance from the present sample w^{n-n_0} . Another method that is often used to capture the most recent behavior of a process is the class of sliding window algorithms in which only a finite-length windowed version of the most recent signal history is used to compute the prediction parameters. Examples of both sliding window and growing window implementations of the

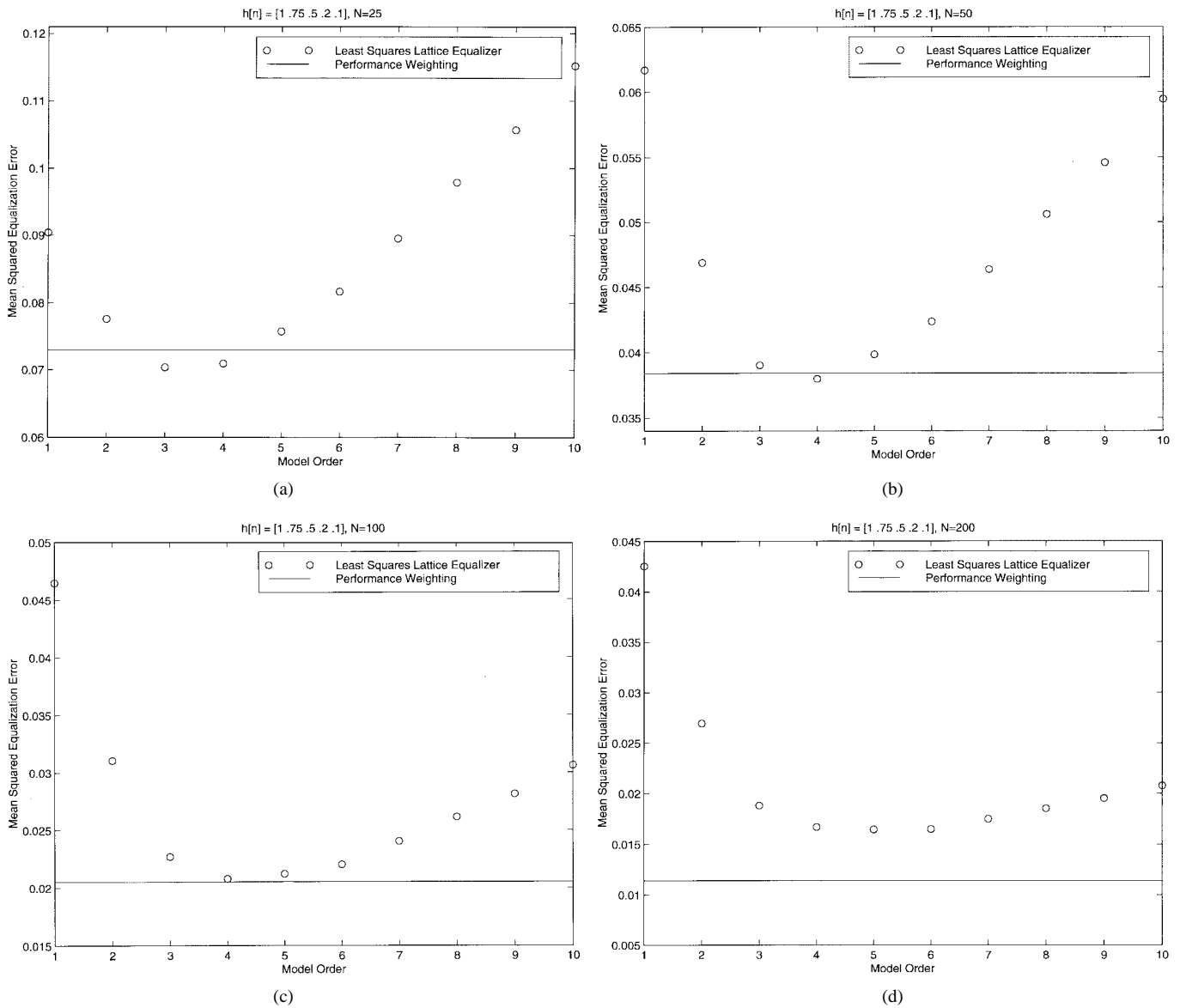


Fig. 8. Mean squared equalization error after training with a BPSK (± 1) sequence of length 25, 50, 100, and 200. Results are shown for an ensemble average over 100 sample training sequences. The average squared equalization error for each of the p th-order linear equalizers, $p = 1, \dots, 10$ are indicated with "o" marks. The average squared equalization error resulting from a performance weighting of all model orders is indicated by the solid line. (a) Twenty-five samples. (b) Fifty samples. (c) One hundred samples. (d) Two hundred samples.

RLS algorithm can be found in [20] and [24]. In Fig. 6, the performance of the universal prediction algorithm of Table I is shown when applied to an autoregressive process that switches between a second- and fourth-order process every 500 samples. The tracking parameter was set to $w = 0.996$, which corresponds to an effective window size of approximately $1/(1-w) = 250$ samples. The top plot in Fig. 6(a) displays the signal to be predicted, which begins as a second-order process and then switches back and forth between a fourth- and second-order process at time sample 500, 1000, and 1500. The middle plot in the figure displays the weights $\mu_2[n]$ and $\mu_4[n]$, which indicate the contribution of the second- and fourth-order predictors to the output of the universal predictor. The bottom plot in the figure indicates the model order whose weight $\mu_k[n]$ was the largest at each point in time. As might be anticipated, initially, it is the first- and then second-order predictor that

has the largest weight. After 500 samples, when the process changes from a second-order to a fourth-order model, the predictor receiving the largest weight becomes the fourth- and third-order models. Once the process changes back to a second-order model after 1000 samples, it is again the second-order predictor that receives the largest weight. Finally, when the process changes back to fourth order, the weight again shifts. Note that although there is a noticeable change in the weights at the transition points, there is finite delay between the process model order change and the time in which the predictor of that order begins to receive a larger weight. In Fig. 6(b), the same set of plots are shown for an algorithm that uses a sliding rectangular window of 250 samples.

Speech processing is a common application in which AR modeling and linear prediction arise [19], [27], [28]. In many applications, an AR model is applied to a segment of speech

over which the signal is assumed to be stationary: typically 10–20 ms. At a sample rate of 10 kHz, for the linear model, we use only on the order of a few hundreds of samples of the speech signal. While there is a tendency to use larger order linear models to extract a finer resolution of the spectral envelope of the speech signal, the larger order models come at cost of temporal resolution since longer segments of speech are required to accurately estimate the parameters of the AR model. This tradeoff between model order and data length, which is pervasive in speech modeling, indicates that our universal approach might be of significant use. As an example, the prediction performance of the universal algorithm of Table I is shown in Fig. 7 for a 10-ms segment from the spoken word “door.” The speech signal was normalized to have unit variance, and the parameter A was set to its maximum absolute value of 2.8. The performance-weighted approach outperforms almost all of the sequential linear predictors as well as the commonly used plug-in approaches.

The final example on data equalization is indicative of the broad scope of adaptive filtering applications to which our performance-weighted approach might apply. To simulate propagation over a multipath channel with a signal to noise ratio of about 30 dB, an ensemble of 100 BPSK (± 1) signals $a[n]$ were convolved with the impulse response of the filter with transfer function $H(z) = 1 + 0.75z^{-1} + 0.5z^{-2} + 0.2z^{-3} + 0.1z^{-4}$ in additive white Gaussian noise of standard deviation 0.025. In Fig. 8, the ensemble average mean squared equalization error after training with 25, 50, 100, and 200 samples are shown. The running-average squared equalization error for each of the p th-order linear equalizers $p = 1, \dots, 10$ are shown along with that resulting from the performance-weighted algorithm of Table II. The universal algorithm rapidly achieves the performance of the best model order and exceeds this performance by the time the data length reaches 100 samples.

VI. CONCLUDING REMARKS

The main result of this paper is an algorithm that is twice universal [8], [9] for linear prediction with respect to model orders and parameters. It uses a performance average prediction of all sequential linear predictors up to some model order. The algorithm is applicable to a variety of signal processing applications, including forecasting, equalization, adaptive filtering, and predictive coding—practically any sequential processing where the measure of performance is the sequentially accumulated mean-square error. The motivating example used in much of this paper was the problem of linear prediction or adaptive AR modeling. Thus, the universal predictor presented here will perform as well as the best linear predictor of any order up to some maximum order uniformly for every bounded individual sequence. As such, the problem of model order selection for linear prediction has been mitigated in favor of a performance-weighted average among all model orders. Since efficient lattice algorithms can be used to recursively generate all of the linear predictors at the computational price of only the largest model order, the universal predictor is computationally very efficient. Extending

the realm of adaptive signal processing algorithms to one in which algorithms are not only optimal in a stochastic framework, but also with respect to individual sequences, is both an exciting direction of this work and an indication that many traditional techniques may be applicable to a much broader class of problems.

ACKNOWLEDGMENT

The authors wish to thank Associate Editor A. H. Sayed and the anonymous reviewers for their detailed comments that considerably improved the clarity of the paper.

REFERENCES

- [1] H. Akaike, “A new look at the statistical model identification,” *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 716–723, Dec. 1974.
- [2] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, pp. 465–471, 1978.
- [3] G. Schwarz, “Estimating the dimension of a model,” *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.
- [4] J. Rissanen, “A predictive least squares principle,” *IMA J. Math. Contr. Inform.*, vol. 3, no. 2-3, pp. 221–222, 1986.
- [5] M. Wax, “Order selection for AR models by predictive least squares,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 581–588, Apr. 1988.
- [6] A. Barron, J. Rissanen, and B. Yu, “The minimum description length principle in coding and modeling (invited paper),” *IEEE Trans. Inform. Theory*, vol. 44, pp. 2743–2760, Oct. 1998.
- [7] N. Merhav and M. Feder, “Universal schemes for sequential decision from individual sequences,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 1280–1292, July 1993.
- [8] B. Y. Ryabko, “Twice-universal coding,” *Probl. Inform. Transm.* vol. 20, pp. 173–177, July/Aug./Sept. 1984.
- [9] ———, “Prediction of random sequences and universal coding,” *Probl. Inform. Transm.*, vol. 24, pp. 87–96, Apr./May/June 1988.
- [10] N. Merhav and M. Feder, “Universal prediction (invited paper),” *IEEE Trans. Inform. Theory*, vol. 44, pp. 2124–2147, Oct. 1998.
- [11] L. D. Davisson, “The prediction error of stationary Gaussian time series of unknown covariance,” *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 527–532, Oct. 1965.
- [12] J. Rissanen, “Universal coding, information, prediction, and estimation,” *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 629–636, 1984.
- [13] M. Feder and A. Singer, “Universal data compression and linear prediction,” in *Proc. 1998 Data Compression Conf.*, Mar. 1998.
- [14] V. Vovk, “Aggregating strategies (learning),” in *Proc. 3rd Annu. Workshop Comput. Learning Theory*, M. Fulk and J. Case, Eds. San Mateo, CA: Morgan Kaufmann, 1990, pp. 371–383.
- [15] P. Elias, “Universal codeword sets and representations of the integers,” *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 194–203, Mar. 1975.
- [16] N. Cesa-Bianchi, Y. Freund, D. Helmbold, D. Haussler, R. Schapire, and M. Warmuth, “How to use expert advice,” in *Proc. Annu. ACM Symp. Theory Comput.*, 1993, pp. 382–391.
- [17] D. Haussler, J. Kivinen, and M. Warmuth, “Tight worst-case loss bounds for predicting with expert advice,” in *Comput. Learning Theory 2nd Euro. Conf.*, P. Vitanyi, Ed., Mar. 1995, pp. 69–83.
- [18] D. T. L. Lee, M. Morf, and B. Friedlander, “Recursive least squares ladder estimation algorithms,” *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 467–481, June 1981.
- [19] M. Morf, A. Vieira, and D. Lee, “Ladder forms for identification and speech processing,” in *Proc. IEEE Conf. Decision Contr.*, Dec. 1977, pp. 1074–1078.
- [20] P. Strobach, “New forms of Levinson and Schur algorithms,” *IEEE Signal Processing Mag.*, pp. 12–36, Jan. 1991.
- [21] A. H. Sayed and T. Kailath, “A state-space approach to adaptive RLS filtering,” *IEEE Signal Processing Mag.*, pp. 18–60, July 1994.
- [22] B. Friedlander, “Lattice methods for spectral estimation,” *Proc. IEEE*, vol. 70, pp. 990–1017, Sept. 1982.
- [23] ———, “Lattice filters for adaptive processing,” *Proc. IEEE*, vol. 70, pp. 829–867, Aug. 1982.
- [24] M. Honig and D. Messerschmidt, *Adaptive Filters: Structures, Algorithms and Applications*. Boston, MA: Kluwer, 1984.
- [25] F. Ling and J. Proakis, “Generalized least squares lattice algorithm and its application to decision feedback equalization,” in *Proc. Int. Conf. Acoust. Speech, Signal Process.*, 1982, vol. 3, pp. 1764–1769.

- [26] E. Satorius and J. Pack, "Application of least squares lattice algorithms to adaptive equalization," *IEEE Trans. Commun.*, vol. COMM-29, pp. 136–142, Feb. 1981.
- [27] J. Makhoul, "Stable and efficient lattice methods for linear prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 423–428, Oct. 1977.
- [28] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ: Prentice-Hall, 1978.



Andrew C. Singer (M'96) received the S.B., S.M., and Ph.D. degrees, all in electrical engineering and computer science, from the Massachusetts Institute of Technology (MIT), Cambridge, in 1990, 1992, and 1996, respectively.

Since 1998, he has been with the faculty of the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, where he is currently an Assistant Professor with the ECE Department and a Research Assistant Professor with the Coordinated Science Laboratory.

During 1996, he was a Postdoctoral Research Affiliate with the Research Laboratory of Electronics at MIT. From 1996 to 1998, he was a Research Scientist with Sanders, a Lockheed Martin Company, Manchester, NH. His research interests include statistical signal processing and communication, universal prediction and data compression, and machine learning.

Dr. Singer is a co-author of the text *Computer Explorations in Signals and Systems* (Englewood Cliffs, NJ: Prentice-Hall). He was a Hughes Aircraft Masters Fellow and was the recipient of the Harold L. Hazen Memorial Award for excellence in teaching in 1991. He is currently a member of the MIT Educational Council and of Eta Kappa Nu and Tau Beta Pi.



Meir Feder (F'99) received the B.Sc and M.Sc. degrees from Tel-Aviv University, Tel-Aviv, Israel, and the Sc.D degree from the Massachusetts Institute of Technology (MIT), Cambridge, and the Woods Hole Oceanographic Institution, Woods Hole, MA, all in electrical engineering in 1980, 1984 and 1987, respectively.

After being a Research Associate and Lecturer with MIT in 1989, he joined the Department of Electrical Engineering—Systems, Tel-Aviv University. He also had visiting appointments at the Woods Hole Oceanographic Institution, Scripps Institute, and Bell Laboratories. From 1995 to 1996, he spend a sabbatical year as a Visiting Professor with the Electrical Engineering and Computer Science Department at MIT. Dr. Feder served as an Associate Editor for Source Coding of the IEEE TRANSACTIONS ON INFORMATION THEORY between June 1993 and June 1996. He received the 1978 "Creative Thinking" Award from the Israeli Defense Forces. He received the 1993 IEEE Information Theory Best Paper Award, the 1994 Tel-Aviv University Prize for Excellent Young Scientists, the 1995 Research Prize of the Israeli Electronic Industry, and in October 1995, he received the Research Prize in Applied Electronics from the Ex-Serviceman Association, London, U.K., which was awarded by Ben-Gurion University.