Universal Piecewise Linear Prediction via Context Trees

Suleyman S. Kozat, Andrew C. Singer and Georg Zeitler

Abstract

This paper considers the problem of piecewise linear prediction from a competitive algorithm approach. In prior work, prediction algorithms have been developed that are "universal" with respect to the class of all linear predictors, such that they perform nearly as well, in terms of total squared prediction error, as the best linear predictor that is able to observe the entire sequence in advance. In this paper, we introduce the use of a "context tree," to compete against a doubly exponential number of piecewise linear (affine) models. We use the context tree to achieve the total squared prediction error performance of the best piecewise linear model that can choose both its partitioning of the regressor space and its real-valued prediction parameters within each region of the partition, based on observing the entire sequence in advance, uniformly, for every bounded individual sequence. This performance is achieved with a prediction algorithm whose complexity is only linear in the depth of the context tree per prediction. Upper bounds on the regret with respect to the best piece-wise linear predictor are given for both the scalar and higher-order case, and lower bounds on the regret are given for the scalar case. An explicit algorithmic description and examples demonstrating the performance of the algorithm are given.

Index Terms

universal, prediction, context tree, piecewise linear.

EDICS Category: MAL-SLER, MAL-PERF, MAL-BAYL

I. Introduction

Linear prediction and linear predictive models have long been central themes within the signal processing literature[1]. More recently, nonlinear models, based on piece-wise linear [27] and locally linear [2] approximations, have gained significant attention as adaptive and Kalman filtering methods also turn to methods such as extended Kalman and particle filtering[3] to capture the salient characteristics of

Suleyman S. Kozat is with IBM, Yorktown, NY, email:kozat@us.ibm.com, Andrew C. Singer and Georg Zeitler are with the Department of ECE at the University of Illinois, Urbana, IL, email:singer@ifp.uiuc.edu.

many physical phenomena. In this paper, we address the problem of sequential prediction and focus our attention on the class of piecewise linear (affine) models. We adopt the nomenclature of the signal processing literature, and use the term "piecewise linear" to refer generally to affine models rather than strictly linear models.

We formulate the prediction problem in a manner similar to that used in machine learning [4], [5], [6], adaptive signal processing [7], [13], and information theory [14], to describe "universal" prediction algorithms, in that they sequentially achieve the performance of the best model from a broad class of models, for every bounded sequence and for a variety of loss functions. These algorithms are sequential such that they may use only the past information, i.e., $x[1], \ldots, x[t-1]$, to predict the next sample $x[t]$. By treating the prediction problem in this context, algorithms are sought which are competitive-optimal with respect to a given class of prediction algorithms, in that they can perform nearly as well, for each and every possible input, as the best predictor that could have been chosen from the competition class, even when this "best predictor" is selected only after observing the entire sequence to be predicted, i.e. non-causally.

Finite and parametrically-continuous linear model classes have been considered, where sequential algorithms that achieve the performance of the best linear model, tuned to the underlying sequence, have been constructed. Competition against linear models is investigated both in prediction and in regression in [4], [7]. However, the structural constraint on linearity considerably limits the potential modeling power of the underlying class, and may be inappropriate for a variety of data exhibiting saturation effects, threshold phenomena, or other nonlinear behavior. As such, the achievable performance of the best linear model may not be a desirable goal in certain scenarios.

In the most general extension of linear models, the prediction is given by an arbitrary nonlinear function, i.e., the prediction of $x[t]$ is given by $f(x[t-1], \ldots, x[1])$ for some arbitrary function f. However, without any constraint on the nonlinear model, this class would be too powerful to compete against, since for any sequence, there always exists a nonlinear function with perfect prediction performance, i.e., one can choose f such that $f(x[t-1], \ldots, x[1]) = x[t]$. By constraining the class of predictors to include piecewise linear (affine) functions, we can retain the breadth of such models, while mitigating the overfitting problems associated with too powerful a competition class. Piecewise linear modeling is a natural nonlinear extension to linear modeling, in which the space spanned by past observations is partitioned into a union of disjoint regions over each of which, an affine model holds. In each region, an estimate of the desired signal is given as the output of a fixed linear regressor. For example, suppose that for a scalar linear predictor, the past observation space, $x[t-1] \in [-A_x, A_x]$ is parsed into J disjoint regions R_j where $\bigcup_{j=1}^{J} R_j = [-A_x, A_x]$ and $A_x \in \mathcal{R}$. At each time t, the underlying predictor forms its prediction of $x[t]$ as $\hat{x}[t] = w_j x[t-1] + c_j$, $w_j \in \mathcal{R}$ and $c_j \in \mathcal{R}$, when $x[t-1] \in R_j$. As the number

of regions grows, the piecewise linear model can better approximate any smoothly varying predictor $\hat{x}[t] = f(x[t-1])$. This statement will be made more precise in the context of the main results of this paper, namely, the upper bounds on regret of the prediction error of the universal predictor can be given with respect to the best piecewise linear prediction algorithm and then extended to bounds on regret with respect to a broad class of smoothly varying nonlinear predictors. Such piecewise linear models have been referred to in the signal processing literature as "nonlinear autoregressive models"[2], and in the signal processing and statistics literature as self-exciting threshold autoregressive (SETAR) models[8], [9], and have been used in modeling a wide range of data in fields ranging from population biology[10] to econometrics[11] to glottal flow in voiced speech[12].

In this paper, we first present results for the piecewise linear regression problem when the regions R_j are fixed and known. We will demonstrate an algorithm that achieves the performance of the best piecewise linear regressor for a given partition and then extend these results to when the boundaries of each region are also design parameters of the class. In this case, we try to achieve the performance of the best sequential piecewise linear regressor when the partitioning of the regressor space is taken from a large class of possible partitions. These partitions will be compactly represented using a "context tree" [17]. Here, we have neither a priori knowledge of the selected partition nor the best model parameters given that partition. We initially focus on scalar piecewise linear regression, such that each prediction algorithm in the competition class is a function of only the latest observation, i.e., $x[t-1]$. These results are then extended to higher-order regression models by considering context tree partitionings of multiple past observations.

We start our discussion when the boundaries of each region are fixed and known. Given such a partition $\bigcup_{j=1}^{J} R_j = [-A_x, A_x]$, the real valued sequence $x^n = \{x[t]\}_{t=1}^n$ is assumed to be bounded but is otherwise arbitrary, in that $|x[t]| < A_x$ for some $A_x < \infty$. Given past values of the desired signal $x[t]$, $t = 1, \ldots, n-1$, we define a competing algorithm from the class of all scalar piecewise affine regressors as

$$\hat{x}_{wc}[t] = w_{s[t-1]} x[t-1] + c_{s[t-1]},$$

where $s[t-1] = j$ when $x[t-1] \in R_j$, $w_j \in \mathcal{R}$ and $c_j \in \mathcal{R}$, $j = 1, \ldots, J$. For each region, $w_j \in \mathcal{R}$ and $c_j \in \mathcal{R}$, $j = 1, \ldots, J$, can be selected independently.

Here we try to minimize the following regret

$$\sup_{x^n} \left\{ \sum_{t=1}^{n} (x[t] - \hat{x}_q[t])^2 - \inf_{\substack{w_j \in \mathcal{R}, c_j \in \mathcal{R} \\ j \in \{1,\ldots,J\}}} \sum_{t=1}^{n} (x[t] - \hat{x}_{wc}[t])^2 \right\}, \tag{1}$$

where, $\hat{x}_{wc}[t] = w_{s[t-1]} y[t] + c_{s[t-1]}$, and $\hat{x}_q[t]$ is the prediction of a sequential algorithm; i.e., we try to achieve the performance of the best model tuned to the underlying sequences x^n.

We first demonstrate an algorithm $\hat{x}[t]$ whose prediction error, over that of the best piecewise linear predictor is upper bounded by $O(2JA_x^2 \ln(n/J))$, i.e,

$$\sum_{t=1}^{n} (x[t] - \hat{x}[t])^2 \leq \inf_{\substack{c_j \in \mathcal{R}, w_j \in \mathcal{R} \\ j \in \{1,\ldots,J\}}} \sum_{t=1}^{n} (x[t] - \hat{x}_{wc}[t])^2 + O(2JA_x^2 \ln(n/J)) \tag{2}$$

for any x^n. Our algorithm pays a "parameter regret" of $O(2A_x^2 \ln(n/J))$ per region to effectively learn (or compete against) the best parameters for that region. We also derive corresponding lower bounds for Equation (1) and show that under certain conditions, our algorithms are optimal in a minmax sense, such that the upper bounds cannot be further improved upon.

We then extend these results and demonstrate an algorithm that achieves the performance of the best sequential predictor (corresponding to a particular partition) from the doubly exponentially large class of such partitioned predictors. To this end, we define a depth-K context tree for a partition with up to 2^K regions, as shown in Figure 1, where, for this tree, $K = 2$. For a depth-K context tree, the 2^K finest partition bins correspond to leaves of the tree. On this tree, each of the bins are equal in size and assigned to regions $[A_x, A_x/2], [A_x/2, 0], [0, -A_x/2], [-A_x/2, -A_x]$. Of course, more general partitioning schemes could be represented by such a context tree.

For a tree of depth-K, there exist $2^{K+1} - 1$ nodes, including leaf nodes and internal nodes. Each node η on this tree represents a portion of the real line, R_η. The region corresponding to each node η, R_η, (if it is not a leaf) is constructed by the union of regions represented by the nodes of its children; the upper node R_{η_u} and the lower node R_{η_l}, $R_\eta = R_{\eta_u} \cup R_{\eta_l}$. By this definition, any inner node is the root of a subtree and represents the union of its corresponding leaves (or bins).

We define a "partition" of the real line as a specific partitioning $\mathcal{P}_i = \{R_{i,1}, \ldots, R_{i,J_i}\}$ with $\bigcup_{j=1}^{J_i} R_{i,j} = [-A_x, A_x]$, where each $R_{i,j}$ is represented by a node on the tree in Figure 1 and $R_{i,j}$ are disjoint. There exist a doubly-exponential number, $N_K \approx (1.5)^{2^K}$, of such partitions, $\mathcal{P}_i$, $i = 1, \ldots, N_K$, embedded within a full depth-K tree. This is equivalent to the number of "proper binary trees" of depth at most K, and is given by Sloane's sequence $A003095$[18], [19]. For each such partition, there exists a corresponding sequential algorithm as in Equation (2) that achieves the performance of the best piecewise affine model for that partition. We can then construct an algorithm that will achieve the performance of the best sequential algorithm from this doubly exponential class.

To achieve the performance of the best sequential algorithm (i.e., the best partition), we try to minimize the following regret

$$\sup_{x^n} \left\{ \sum_{t=1}^{n} (x[t] - \hat{x}_q[t])^2 - \inf_{\mathcal{P}_i} \sum_{t=1}^{n} \left(x[t] - \hat{x}_{\mathcal{P}_i}[t] \right)^2 \right\}, \tag{3}$$

where $\hat{x}_{\mathcal{P}_i}[t]$ is the corresponding sequential piecewise linear predictor for partition $\mathcal{P}_i$, and $\hat{x}_q[t]$ is the prediction of a sequential algorithm.

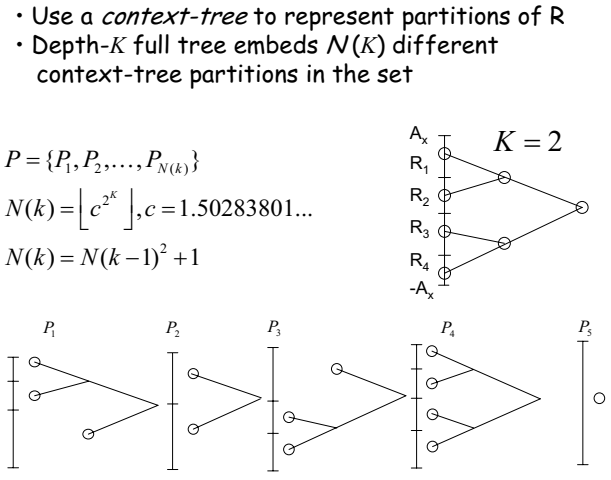


Fig. 1. A full tree of depth 2 that represents all context-tree partitions of the real line $[-A_x, A_x]$ into at most four possible regions.

We will then demonstrate a sequential algorithm, $\hat{x}[t]$, such that the "structural regret" in Equation (3) is at most $O(2C(\mathcal{P}_i))$, where $C(\mathcal{P}_i)$ is a constant which depends only on the partition $\mathcal{P}_i$, i.e.,

$$\sum_{t=1}^{n} (x[t] - \hat{x}[t])^2 \leq \inf_{\mathcal{P}_i} \sum_{t=1}^{n} \left(x[t] - \hat{x}_{\mathcal{P}_i}[t] \right)^2 + O(2C(\mathcal{P}_i)),$$

which yields, upon combining the parameter and structural regret, an algorithm achieving

$$\sum_{t=1}^{n} (x[t] - \hat{x}[t])^2 \leq \inf_{\mathcal{P}_i} \left\{ \inf_{w_{i,j}, c_{i,j} \in \mathcal{R}} \sum_{t=1}^{n} \left(x[t] - \hat{x}_{\mathcal{P}_i}[t] \right)^2 + O(2J \ln(n/J)) + O(2C(\mathcal{P}_i)) \right\},$$

uniformly for any x^n, where $\hat{x}_{\mathcal{P}_i}[t] = w_{i,s_i[t-1]} x[t-1] + c_{i,s_i[t-1]}$.

Hence, the algorithms introduced here are "twice-universal" in that they asymptotically achieve the prediction performance of the best predictor in which the regression parameters of the piecewise linear model and also the partitioning structure of the model itself can be selected based on observing the whole sequence in advance. Our approach is based on sequential probability assignment from universal source coding [17], [23], [24] and uses the notion of a context tree from [17] to compactly represent the N_K partitions of the regressor space. Here, instead of making hard decisions at each step of the algorithm to select a partition or its local parameters, we use a soft combination of all possible models and parameters to achieve the performance of the best model, with complexity that remains linear in the depth of the context tree per prediction.

In [14], sequential algorithms based on "plug-in" predictors are demonstrated that approach the best batch performance with additional regret of $O(n^{-1} \ln(n))$ with respect to certain nonlinear prediction

classes that can be implemented by finite-state machines. It is shown that Markovian predictors with sufficiently long memory are asymptotically as good as any given finite-state predictor for finite-alphabet data. A similar problem is investigated for online prediction for classes of smooth functions in [6], where corresponding upper and matching lower bounds are found (in some cases) when there is additional information about the data, such as the prediction error of the best predictor for a given sequence. The problem of universal nonlinear prediction has also been investigated in a probabilistic context. In [25] the authors propose a universal minimum complexity estimator for the conditional mean (minimum mean square error predictor) of a sample, given the past, for a finite memory process, without knowing the true order of the process. In [15], a class of elementary universal predictors of an unknown nonlinear system is considered using an adaptation of the well-known nearest neighbor algorithm [26]. They are universal in the sense that they predict asymptotically well for every bounded input sequence, every disturbance sequence in certain classes, and every nonlinear system, given certain regularity conditions. In [27], a regression tree approach is developed for identification and prediction of signals that evolve according to an unknown nonlinear state space model. Here a tree is recursively constructed partitioning the state space into a collection of piecewise homogeneous regions, resulting a predictor which nearly attains the minimum mean squared error.

In the computational learning theory literature, the related problem of prediction as well as the best pruning of a decision tree has been considered, in which data structures and algorithms similar to context trees have been used[20], [21], [22]. In [20] the authors develop a sequential algorithm that, given a decision tree, can nearly achieve the performance of the best pruning of that decision tree, under the absolute loss function. While the data structure used is similar to that we develop, its use is similar in spirit to that of the Willems, et al. context-tree weighting paper [17], in which the "context" of the data sequence is based on a temporal parsing of the binary data sequence. As such, the leaves of a given context tree (or pruning of the decision tree) are reached at depth k after k symbols of the data have been observed. The predictor then makes its prediction based on the label assigned to the leaf of the tree reached by the sequence. In [20], the observed sequence is binary, i.e., $x^n \in \{0,1\}^n$, while the predictions are real-valued, $\hat{x}^n \in [0,1]^n$, but fixed for each leaf of the decision tree.

These results are extended to other loss functions, including the square error loss, in [21], [22] using similar methods to [20] and an approach based on dynamic programming. The main result of [21] is an algorithm that competes well against all possible prunings of a given decision tree and upper bounds on the regret with respect to the best pruning. However, in this result, predictions are permitted only to be given by the labels of the leaves of the decision tree. As such, the main result of [21] considers essentially competing against a finite (albeit large) number of predictor models. While the label function is permitted to change with time (time-varying predictors at each leaf), it is only in the last section of

[21] that the competition class of predictor models is extended to include all possible labellings of the leaves of the tree. However, for this case, the discussion and the subsequent bounds are only given for binary sequences x^n, for finite-alphabet predictions $\hat{x}[n]$ and the absolute loss function, rather than the continuous-alphabet quadratic loss problem discussed in this paper. In our work, we consider piecewise linear predictors, which would correspond to labels within the leaves of the pruned decision tree that are not single prediction values (labels), but are rather *functions* of samples of the sequence x^n, i.e. the regressor space, $x[n-1], x[n-2], \ldots x[n-p]$. Further, the "context" used in our context trees correspond to a *spatial* parsing of the regressor space, rather than the *temporal* parsing discussed in [21], [22], [20]. Another key difference between this related work and that developed here, is the constructive nature of our results. We illustrate a prediction algorithm with a time complexity that is linear in the depth of the context tree and whose algebraic operations are explicitly given in the text. This is in contrast to the methods described in these related works, whose time complexity is stated as polynomial (in some cases linear), but whose explicit algebraic description is not completely given. This is largely due to the search-like process necessary to carry out the final prediction step in the aggregating algorithm, on which these methods build.

We begin our discussion of piecewise linear modeling with the case when the partition is fixed and known in Section II. We then extend these results using context trees in Section III to include comparison classes with arbitrary partitions from a doubly exponential class of possible partitions. In each section, we provide theorems that upper-bound the regret with respect to the best competing algorithm in the class. The theorems are constructive, in that they yield algorithms satisfying the corresponding bounds. An explicit MATLAB implementation of the context tree prediction algorithm is also given. Extensions to higher-order piecewise linear prediction algorithms are given in Section IV. Lower bounds on the achievable regret are discussed in Section V. The paper is then concluded with simulations of the algorithms on synthetic and real data.

II. Piecewise Linear Prediction: Known Regions

In this section, we consider the problem of predicting as well as the best piecewise affine predictor, when the partition of the regression space is given and known. As such, we seek to minimize the following regret

$$\sup_{x^n} \left\{ \sum_{t=1}^{n} \left(x[t] - \hat{x}_q[t]\right)^2 - \inf_{w_j \in \mathcal{R}, c_j \in \mathcal{R}} \sum_{t=1}^{n} \left(x[t] - w_{s[t-1]} x[t-1] - c_{s[t-1]}\right)^2 \right\}, \tag{4}$$

where $\hat{x}_q[t]$ is the prediction from a sequential algorithm and $|x[t]| \leq A_x$. That is, we wish to obtain a sequential algorithm that can predict every sequence x^n as well as the best fixed piecewise linear (affine) algorithm for that sequence with a partition of the regressor space given by $\bigcup_{j=1}^{J} R_j = [-A_x, A_x]$.

One of the predictors from the class against which this algorithm will compete can represented by the parameter vector $\vec{\theta} = [c_1, \ldots, c_J, w_1, \ldots, w_J]^T$ and would accumulate the loss

$$l_n(x, \hat{x}_{\vec{\theta}}) \stackrel{\triangle}{=} \sum_{t=1}^{n} (x[t] - w_{s[t-1]} x[t-1] - c_{s[t-1]})^2. \tag{5}$$

Equation (5) can be written in a more compact form if we define extended vectors, $\vec{y}[t] = [x[t-1]\ 1]^T$ and $\vec{w}_{s[t-1]}[t] = [w_{s[t-1]}\ c_{s[t-1]}]^T$,

$$l_n(x, \hat{x}_{\vec{\theta}}) = \sum_{t=1}^{n} (x[t] - \vec{w}^T[t]\vec{y}[t])^2.$$

Since the number and boundaries of the regions are known, we have J independent least squares problems. Defining J time vectors (or index sequences) of length n_j, $t_j^{n_j} = \{t : s[t-1] = j\}$, with $j = 1, \ldots, J$, and sequences $x_j^{n_j} = \{x[t_j[k]]\}_{k=1}^{n_j}$ and $\vec{y}_j^{n_j} = \{\vec{y}[t_j[k]]\}_{k=1}^{n_j}$, then the universal predictor we suggest can be constructed using the universal affine predictor of [7] in each region, i.e.,

$$\tilde{x}_{\vec{w}}[n] = \tilde{\vec{w}}_{s[n-1]}^T[n-1]\vec{y}[n]$$

with

$$\tilde{\vec{w}}_j[n-1] = (D_{\vec{y}_j\vec{y}_j}^{n_j} + \delta_j I)^{-1} D_{x_j\vec{y}_j}^{n_j - 1}, \tag{6}$$

where n_j is the number of points of x^{n-1} that belong to R_j, $\delta_j > 0$ is a positive constant, $D_{x_j\vec{y}_j}^{n_j-1} = \sum_{t=1}^{n_j-1} x[t_j[t]]\vec{y}[t_j[t]]$, $D_{\vec{y}_j\vec{y}_j}^{n_j} = \sum_{t=1}^{n_j} \vec{y}[t_j[t]]\vec{y}^T[t_j[t]]$ and I is an appropriate sized identity matrix. The following theorem relates the performance of the universal predictor, $l_n(x, \tilde{x}_{\vec{w}}) = \sum_{t=1}^{n} (x[t] - \tilde{x}_{\vec{w}}[t])^2$, to that of the best batch scalar piecewise linear predictor.

Theorem 1: *Let x^n be an arbitrary bounded, real-valued sequence, such that $|x[t]| < A_x$ for all t. Then $l_n(x, \tilde{x}_{\vec{w}})$ satisfies*

$$\frac{1}{n} l_n(x, \tilde{x}_{\vec{w}}) \leq \frac{1}{n} \min_{\vec{\theta}} \{l_n(x, \hat{x}_{\vec{\theta}}) + \delta \|\vec{\theta}\|^2\} + \frac{1}{n} \sum_{j=1}^{J} 2h_j \ln\left(1 + \frac{n_j A_x^2}{\delta_j}\right) \tag{7}$$

with

$$h_j = \frac{1}{n_j} \sum_{k=1}^{J} n_{jk} A_{x,k}^2,$$

where n_{jk} is the number of elements of region k that result from a transition from region j and $|x[t]| \leq A_{x,k}$ when $x[t] \in R_k$,, $\delta > 0$ and $\delta_j > 0$ are arbitrary constants.

Here, $l_n(x, \hat{x}_{\vec{\theta}}) = \sum_{t=1}^{n} (x[t] - w_{s[t-1]} x[t-1] - c_{s[t-1]})^2$ and $s[t-1]$ is the state indicator variable.

The proof of Theorem 1 is based on sequential probability assignment and follows directly from [28]. A relaxed, but perhaps more straightforward upper bound on the right hand side of (7) can be obtained by maximizing the upper bound with respect to n_j, replacing $A_{x,k}$ with A_x and δ_j with δ yields

$$\frac{1}{n} l_n(x, \tilde{x}_{\vec{w}}) - \frac{1}{n} \min_{\vec{\theta}} \{l_n(x, \hat{x}_{\vec{\theta}}) + \delta \|\vec{w}\|^2\} \leq 2JA_x^2 \frac{\ln(n/J)}{n} + O\left(\frac{1}{n}\right). \tag{8}$$

III. Piecewise Linear Prediction: Context Trees

We now consider the prediction problem where the class against which the algorithm must compete includes not only the best predictor for a given partition, but also the best partition of the regressor space as well. As such, we are interested in the following regret

$$\sup_{x^n} \left\{ \sum_{t=1}^{n} \left(x[t] - \hat{x}_q[t]\right)^2 - \inf_{\mathcal{P}_i} \sum_{t=1}^{n} \left(x[t] - \hat{x}_{\mathcal{P}_i}[t]\right)^2 \right\},$$

where $\hat{x}_q[t]$ is the prediction of any sequential algorithm, $\mathcal{P}_i$ is a partition of the real line with the state indicator variable $s_i[t-1] = j$ if $x[t-1] \in R_{i,j}$, and $\mathcal{P}_i = \{R_{i,1}, \ldots, R_{i,J_i}\}$ with $\bigcup_{j=1}^{J_i} R_{i,j} = [-A_x, A_x]$ for some J_i, and $\hat{x}_{\mathcal{P}_i}[t]$ is the corresponding sequential algorithm for the partition $\mathcal{P}_i$. The partition $\mathcal{P}_i$ can be viewed as in Figure 1 as a subtree or "context tree" of a depth K full tree with the $R_{i,j}$ corresponding to the nodes of the tree. Each $R_{i,j}$ is represented by a node on the full tree and $R_{i,j}$ are disjoint. Given the full tree, there exist N_K such partitions, i.e., $\mathcal{P}_i$, $i = 1, \ldots, N_K$, where $N_K = N_{K-1}^2 + 1$. Although, we use the sequential predictors introduced in Theorem 1 for each partition $\mathcal{P}_i$, our algorithm has no such restrictions; given any sequential algorithms running independently within each region $R_{i,j}$, our algorithm will achieve the performance of the best partition with the corresponding sequential algorithms. Nevertheless, by using these specific universal algorithms in each region, we also achieve the performance of the best affine model for that region from the continuum of all affine predictors for any bounded data x^n. Hence, our algorithms are twice-universal [30].

Similar to [24], we define $C(\mathcal{P}_i)$ as the number of bits that would have been required to represent each partition $\mathcal{P}_i$ on the tree using a universal code:

$$C(\mathcal{P}_i) = J_i + n_{\mathcal{P}_i} - 1,$$

where $n_{\mathcal{P}_i}$ is the total number of leaves in $\mathcal{P}_i$ that have depth less than K, i.e., leaves of $\mathcal{P}_i$ that are inner nodes of the tree. Since $n_{\mathcal{P}_i} \leq J_i$,

$$C(\mathcal{P}_i) \leq 2J_i - 1.$$

We note that for our context tree, this definition of $C(\mathcal{P}_i)$ is identical to the "size" of a pruning $|\mathcal{P}_i|$ used in [20]. Given the tree, we can construct a sequential algorithm with linear complexity in the depth of the context tree per prediction that asymptotically achieves both the performance of the best sequential predictor and also the performance of the best affine predictor for any partition as follows.

Theorem 2: *Let x^n be an arbitrary bounded scalar real-valued sequence, such that $|x[t]|| < A_x$, for all t. Then we can construct a sequential predictor $\tilde{x}_{wlin}[t]$ with complexity linear in the depth of the context*

tree per prediction such that

$$\sum_{t=1}^{n}(x[t]-\tilde{x}_{wlin}[t])^2 \leq \inf_{\mathcal{P}_i}\left(\inf_{w_{i,j}\in\mathcal{R},c_{i,j}\in\mathcal{R}}\left\{\sum_{t=1}^{n}\left(x[t]-\vec{w}_{i,s_i[t-1]}^T\vec{y}[t]\right)^2+\delta(\|\vec{w}_i\|^2)\right\}+ \right. \quad (9)$$
$$\left. 8A_x^2C(\mathcal{P}_i)\ln(2)+2J_iA_x^2\ln(n/J_i)\right)+O(1),$$

and also another sequential predictor $\tilde{x}_{wpol}[t]$ with complexity polynomial in the depth of the context tree per prediction such that

$$\sum_{t=1}^{n}(x[t]-\tilde{x}_{wpol}[t])^2 \leq \inf_{\mathcal{P}_i}\left(\inf_{w_{i,j}\in\mathcal{R},c_{i,j}\in\mathcal{R}}\left\{\sum_{t=1}^{n}\left(x[t]-\vec{w}_{i,s_i[t-1]}^T\vec{y}[t]\right)^2+\delta(\|\vec{w}_i\|^2)\right\}+\right.$$
$$\left. 2A_x^2C(\mathcal{P}_i)\ln(2)+2J_iA_x^2\ln(n/J_i)\right)+O(1),$$

where $\delta>0$, $\mathcal{P}_i$ is any partition on the context tree, $C(\mathcal{P}_i)$ is a constant that is less than or equal to $2J_i-1$, $\vec{w}_{i,s_i[t-1]}=[w_{i,s_i[t-1]}\ \ c_{i,s_i[t-1]}]^T$, $\vec{y}[t]=[x[t-1]\ \ 1]^T$.

The construction of the universal predictor $\tilde{x}_{wlin}[t]$ and $\tilde{x}_{wpol}[t]$ are given at the end of the proof of Theorem 2. Note that the inequality in Theorem 2 holds for *any* partition of the data, including that achieving $\inf_{\mathcal{P}_i}$ over the right hand side. This implies that, without prior knowledge of any complexity constraint on the algorithm, such as prior knowledge of the depth of the context tree against which it is competing, the universal prediction algorithm can compete well with each and every subpartition (context-tree) within the depth-K full tree used in its construction.

In the derivation of the universal algorithm we observe the following result. Suppose we are given sequential predictors, $\hat{x}_\eta[t]$, for each node η on the context tree. Without any restriction on these sequential algorithms, $\hat{x}_\eta[t]$, we have the following theorem:

Theorem 3: *Let x^n be an arbitrary bounded scalar real-valued sequence, with $|x[t]|<A_x$ for all t. Given a context tree with corresponding nodes η, $\eta=\{1,\ldots,2^{K+1}-1\}$ and sequential predictors for each node $\hat{x}_\eta[t]$, we can construct a sequential predictor $\tilde{x}_{wlin}[t]$ with complexity linear in the depth of the context tree per prediction such that*

$$\sum_{t=1}^{n}(x[t]-\tilde{x}_{wlin}[t])^2 \leq \inf_{\mathcal{P}_i}\left(\sum_{t=1}^{n}\left(x[t]-\hat{x}_{\mathcal{P}_i}[t]\right)^2+8A_x^2C(\mathcal{P}_i)\ln(2)\right)+O(1)$$

and another sequential predictor $\tilde{x}_{wpol}[t]$ with complexity polynomial in the depth of the context tree per prediction such that

$$\sum_{t=1}^{n}(x[t]-\tilde{x}_{wpol}[t])^2 \leq \inf_{\mathcal{P}_i}\left(\sum_{t=1}^{n}\left(x[t]-\hat{x}_{\mathcal{P}_i}[t]\right)^2+2A_x^2C(\mathcal{P}_i)\ln(2)\right)+O(1)$$

where $\delta>0$ and $C(\mathcal{P}_i)$ is a constant that is less than or equal to $2J_i-1$, and $\hat{x}_{\mathcal{P}_i}[t]$ is the sequential predictor obtained by the combination of the sequential predictors corresponding to its piecewise regions

$\mathcal{P}_i = \{R_{i,1}, \ldots, R_{i,J}\}$, *i.e.,* $\hat{x}_{\mathcal{P}_i}[t] = \hat{x}_\eta[t]$ *if* $x[t-1] \in R_{i,j}$ *and* $R_{i,j} = R_\eta$.

We can also consider expanding the class of predictors against which the algorithm must compete to include any smooth nonlinear function f in the following manner.

Corollary 3: *Let x^n be an arbitrary bounded scalar real-valued sequence, with $|x[t]| < A_x$ for all t. Let $\mathcal{F}$ be the class of all twice differentiable functions $\hat{x}[t] = f(x[t-1])$, such that $|f_{xx}| < K_2$, $K_2 \geq 0$, where f_{xx} is the second derivative of f. Then we have that $\hat{x}_K[t]$ satisfies,*

$$\sum_{t=1}^{n}(x[t] - \hat{x}_K[t])^2 \leq \inf_{f\in\mathcal{F}} \sum_{t=1}^{n} \left(x[t] - f(x[t-1])\right)^2 + \frac{n}{2}K_2 2^{-2K} + O(\ln(n))$$

where, $\hat{x}_K[t]$ is a depth-K context-tree predictor of Theorem 2.

Corollary 3 follows from Theorems 2 and 3 and application of the Lagrange form of Taylor's theorem applied to f about the midpoint of each region in the finest partition.

A. Proof of Theorem 2

We first prove Theorem 2, for piecewise constant models, i.e., when $\vec{y}[t] = [0\ 1]^T$ and $\hat{x}_w[t] = c_{s[t-1]}$, $c_{s[t-1]} \in \mathcal{R}$. The proof will then be extended to include affine models.

Given a partition $\mathcal{P}_i = \bigcup_{j=1}^{J_i} R_{i,j}$, we consider a family of predictors, $\mathcal{P}_i \in \mathcal{P}$ (the competing class) each with its own prediction vector $\vec{c}_i = [c_{i,1}, \ldots, c_{i,J_i}]^T$. Here, each $c_{i,j}$ represents a constant prediction for the jth region of partition $\mathcal{P}_i$, i.e., when $x[t-1] \in R_{i,j}$, $\hat{x}_{\vec{c}_i} = c_{i,j}$. For each pairing of $\mathcal{P}_i$ and $\vec{c}_i$, we also consider a measure of the sequential prediction performance, or loss, of the corresponding algorithm,

$$l_n(x, \hat{x}_{\vec{c}_i}|\vec{c}_i, \mathcal{P}_i) \stackrel{\triangle}{=} \sum_{t=1}^{n}(x[t] - c_{i,s_i[t-1]})^2,$$

where $s_i[t-1]$ is the state indicator variable for partition $\mathcal{P}_i$, i.e., $s_i[t-1] = j$ if $x[t-1] \in R_{i,j}$. We define a function of the loss, namely, the "probability"

$$\begin{aligned} P(x^n \mid \vec{c}_i, \mathcal{P}_i) &\stackrel{\triangle}{=} \exp\left(-\frac{1}{2a}\sum_{t=1}^{n}(x[t] - c_{i,s_i[t-1]})^2\right) \\ &= \exp\left(-\frac{1}{2a} l_n(x, \hat{x}_{\vec{c}_i}|\vec{c}_i, \mathcal{P}_i)\right), \end{aligned}$$

which can be viewed as a probability assignment of $\mathcal{P}_i$, with parameters $\vec{c}_i$, to x^n induced by the performance of the corresponding predictor with $\mathcal{P}_i$ and $\vec{c}_i$ on the sequence x^n, where a is a positive constant, related to the learning rate of the algorithm. Given $\mathcal{P}_i$, the algorithm in the family with the best constant predictor in each region assigns to x^n the probability

$$P^*(x^n \mid \mathcal{P}_i) \stackrel{\triangle}{=} \exp\left(-\frac{1}{2a}\inf_{\vec{c}_i} l_n(x, \hat{x}_{\vec{c}_i}|\vec{c}_i, \mathcal{P}_i)\right).$$

Maximizing $P^*(x^n|\mathcal{P}_i)$ over all $\mathcal{P}_i$ (on the tree) yields

$$P^*(x^n|\mathcal{P}_i{}^*) \stackrel{\triangle}{=} \sup_{\mathcal{P}_i} P^*(x^n|\mathcal{P}_i).$$

Here, $P^*(x^n|\mathcal{P}_i{}^*)$ corresponds to the best piecewise constant predictor in the class on the tree of depth K. Note that without any constraint on the complexity of the algorithms in the competing class, and since this performance is computed based on observation of the entire sequence in advance, $P^*(x^n|\mathcal{P}_i{}^*)$ would correspond to the finest partition of the interval with the best piecewise constant predictors in each interval, i.e., the full binary tree. There is no guarantee, however, that the sequential performance of our algorithm will be the best if we choose the finest-grain model, given the increase in the number of parameters that must be learned sequentially by the algorithm, and, correspondingly, the increase in the regret with respect to the best "batch" algorithm, which is permitted to select all of its parameters in hindsight (i.e., given all of the data in advance). In fact, the finest grain model generally will not have the best performance when the algorithms are required to sequentially compete with the best batch algorithm within each partition. As such, our goal is to perform well with respect to all possible partitions. As will be shown, the context-tree weighting approach enables the algorithm to achieve the performance of the best partition-based algorithm. Within each partition, the algorithm sequentially achieves the performance of the best batch algorithm. This "twice-universality," once over the class of partitions of the regressor space, and again over the set of parameters within each partition, enables the algorithm to sequentially achieve the best possible performance out of the doubly exponential number, N_K, of partitions and the infinite set of parameters given the partition.

Given any $\mathcal{P}_i$, using the sequential algorithm introduced in Equation (6) with $\tilde{c}_j[n] = \tilde{x}_{\vec{w}}[n]$, where $\vec{w}_j = [0\, c_j]$ and $\vec{y}[t] = [0\, 1]^T$, for all t, and for the partition $\mathcal{P}_i$ yields

$$\tilde{P}(x^n|\mathcal{P}_i) \stackrel{\triangle}{=} \exp\left(-\frac{1}{2a}\sum_{t=1}^{n}(x[t] - \tilde{c}_{s_i[t-1]}[t-1])^2\right). \tag{10}$$

As the first step, we will derive a universal probability assignment, $\tilde{P}_u(x^n)$, to x^n as a weighted combination of probabilities on the context tree. We will then demonstrate that this universal probability is asymptotically as large as that of any predictor in the class, including $P^*(x^n|\mathcal{P}_i{}^*)$. As the final step we construct a sequential prediction algorithm of linear complexity whose associated probability assignment to x^n is as large as $\tilde{P}_u(x^n)$ and hence the desired result.

As the next step, we assign to each node η on the context tree a sequential predictor working only on the data observed by this particular node. For a node η representing the region R_η, we first assign a time vector (or index sequence) of length n_η, $t_\eta^{n_\eta} = \{t : x[t-1] \in R_\eta\}$ and a sequence $d_\eta^{n_\eta} = \{x[t_\eta^{n_\eta}[k]]\}_{k=1}^{n_\eta}$. Clearly, for each node η, there corresponds a portion of the observation sequence of length n_η and for a parent node in the tree with upper and lower children we have $n_\eta = n_{\eta_u} + n_{\eta_l}$, where n_{η_u} is the

length of the subsequence that is shared with the upper child and n_{η_l} is the partition shared with the lower child. For each node, we assign a predictor

$$\tilde{c}_\eta[n] = \frac{\sum_{t=1}^{n\eta} d_\eta[t]}{n\eta + 1 + \delta} \tag{11}$$

where δ is a positive constant for the prediction of $d_\eta[n\eta + 1]$.

We then define a weighted probability of a leaf node as

$$\tilde{P}_\eta(x^n) = \exp\left(-\frac{1}{2a}\sum_{t=1}^{n\eta}(d_\eta[t] - \tilde{c}_\eta[t-1])^2\right), \tag{12}$$

which is a function of the performance of the node predictor on the sequence $d_\eta^{n\eta}$. The probability of an inner node is defined as [17]

$$\tilde{P}_\eta(x^n) = \frac{1}{2}\tilde{P}_{\eta_u}(x^n)\tilde{P}_{\eta_l}(x^n) + \frac{1}{2}\exp\left(-\frac{1}{2a}\sum_{t=1}^{n\eta}(d_\eta[t] - \tilde{c}_\eta[t-1])^2\right), \tag{13}$$

which is a weighted combination of the probabilities assigned to the data by each of the child nodes operating on the substrings, $x[t_{\eta_u}^{n_{\eta_u}}]$ and $x[t_{\eta_l}^{n_{\eta_l}}]$, $\tilde{P}_{\eta_u}(x^n)$ and $\tilde{P}_{\eta_l}(x^n)$, and the probability assigned to $d_\eta^{n\eta}$ by the sequential predictor of R_η. We then define the universal probability $\tilde{P}_u(x^n)$ of x^n as the probability of the root node

$$\tilde{P}_u(x^n) = \tilde{P}_r(x^n),$$

where we represent the root note with $\eta = r$. Using the recursion in Equation (13), it can be shown, as in Lemma 2 of [24], that the root probability $\tilde{P}_r(x^n)$ is given by the sum of weighted probabilities of partitions $\mathcal{P}_i$

$$\tilde{P}_u(x^n) = \sum_{\mathcal{P}_i} 2^{-C(\mathcal{P}_i)}\tilde{P}(x^n|\mathcal{P}_i),$$

where $C(\mathcal{P}_i) = J_i + n_{\mathcal{P}_i} - 1$ is defined as the "cost" of partition $\mathcal{P}_i$ and $P(\mathcal{P}_i) \stackrel{\triangle}{=} 2^{-C(\mathcal{P}_i)}$ can be viewed as a prior weighting of the partition $\mathcal{P}_i$. It can also be shown that $\sum_{\mathcal{P}_i} 2^{-C(\mathcal{P}_i)} = 1$ [17].

Hence, for any $\mathcal{P}_i$

$$\tilde{P}_u(x^n) \geq 2^{-C(\mathcal{P}_i)}\tilde{P}(x^n|\mathcal{P}_i),$$

since $\mathcal{P}_i \geq 0$ and $\tilde{P}(x^n|\mathcal{P}_i) \geq 0$, for all i, this yields

$$-2a\ln(\tilde{P}_u(x^n)) \leq 2aC(\mathcal{P}_i)\ln(2) - 2a\ln(\tilde{P}(x^n|\mathcal{P}_i)).$$

Using Equation (8) on $\tilde{P}(x^n|\mathcal{P}_i)$, we obtain

$$-2a\ln(\tilde{P}_u(x^n)) \leq \tag{14}$$
$$2aC(\mathcal{P}_i)\ln(2) + \inf_{c_{i,j}\in\mathcal{R}}\left\{\sum_{t=1}^{n}\left(x[t] - c_{i,s_i[t-1]}\right)^2 + \delta\|\vec{c}_i\|^2\right\} + J_i A_x^2\ln(n/J_i) + O(1).$$

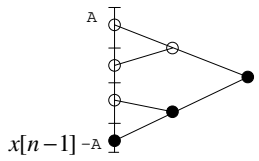


Fig. 2. $K+1$ nodes to be updated.

Hence we have a probability assignment $\tilde{P}_u(x^n)$ which is as large as the probability assignment of the best partition $P^*(x^n|\mathcal{P}_i{}^*)$ to x^n, to first order in the exponent. However, $\tilde{P}_u(x^n)$ is not in the form of the assigned probability from a valid sequential predictor. That is, we have no prediction algorithm that achieves $\tilde{P}_u(x^n)$. We now demonstrate a sequential prediction algorithm whose probability assignment to x^n is as large as $\tilde{P}_u(x^n)$ and which is also in the proper prediction form, i.e. it arises from a valid sequential predictor.

The universal probability $\tilde{P}_u(x^n)$ can be calculated recursively by defining a conditional probability, from the induced probability, i.e.,

$$\tilde{P}_u(x[n]|x^{n-1}) \stackrel{\triangle}{=} \frac{\tilde{P}_u(x^n)}{\tilde{P}_u(x^{n-1})},$$

where $\tilde{P}_u(x^n) = \prod_{t=1}^n \tilde{P}_u(x[t]|x^{t-1})$. To achieve $\tilde{P}_u(x^n)$, we will demonstrate a sequential algorithm with probability assignment as large or larger than $\tilde{P}_u(x[t]|x^{t-1})$ for all t. For this, we will present a sequential update from $\tilde{P}_u(x^{n-1})$ to $\tilde{P}_u(x^n)$.

Given x^{n-1} and $\tilde{P}_u(x^{n-1})$, node probabilities $P_\eta(x^{n-1})$ should be adjusted after observing $x[n]$ to form $\tilde{P}_u(x^n)$. However, owing to the tree structure, only probabilities of nodes that include $x[n-1]$ need to be updated to form $\tilde{P}_u(x^n)$. We have $K+1$ nodes that contain $x[n-1]$: the leaf node that contains $x[n-1]$ and all the nodes that contain the leaf that contains $x[n-1]$. Hence, at each time n, only $K+1$ node probabilities in $\tilde{P}_u(x^{n-1})$ must be adjusted to form $\tilde{P}_u(x^n)$. This enables us to update $\tilde{P}_u(x^{n-1})$, a mixture of all N_K predictors with only $K+1$ updates, instead of updating all $N_K \approx (1.5)^{2^K}$ predictor probabilities to reach $\tilde{P}_u(x^n)$.

We now illustrate this update procedure by an example. Without loss of generality, suppose $x[n-1]$ belongs to the lowest leaf of the tree in Figure 2. All the nodes along the path of nodes indicated by filled circles in Figure 2 include $x[n-1]$ and only these need to be updated after observing $x[n]$. For any $x[n-1]$ there exits such a path of $K+1$ nodes, which we refer to as "dark nodes.". Here, we represent the root node as $\eta = r$; upper and lower children of the root node as r_u and r_l; and recursively, the upper child of the upper child of the root node as r_{uu} and the lower child of the upper child of the parent node as r_{ul}. By this notation, we will now apply the recursion in Equation (13) to all dark nodes in $\tilde{P}_u(x^{n-1})$,

and indicate those probabilities updated using the symbol "$\Downarrow$" in Equation (15), to obtain

$$\tilde{P}_u(x^{n-1}) = \frac{1}{2}\tilde{P}_{r_u}(x^{n-1})\overset{\Downarrow}{\tilde{P}}_{r_l}(x^{n-1}) + \frac{1}{2}\exp\left(-\frac{1}{2a}\sum_{t=1}^{n_r-1}(d_r[t]-\tilde{c}_r[t-1])^2\right), \tag{15}$$

$$= \frac{1}{2}\tilde{P}_{r_u}(x^{n-1})\left(\frac{1}{2}\tilde{P}_{r_{lu}}(x^{n-1})\overset{\Downarrow}{\tilde{P}}_{r_{ll}}(x^{n-1}) + \frac{1}{2}\exp\left(-\frac{1}{2a}\sum_{t=1}^{n_{r_l}-1}(d_{r_l}[t]-\tilde{c}_{r_l}[t-1])^2\right)\right)$$

$$+ \frac{1}{2}\exp\left(-\frac{1}{2a}\sum_{t=1}^{n_r-1}(d_r[t]-\tilde{c}_r[t-1])^2\right),$$

$$= \frac{1}{2}\tilde{P}_{r_u}(x^{n-1})\left(\frac{1}{2}\tilde{P}_{r_{lu}}(x^{n-1})\frac{1}{2}\exp\left(-\frac{1}{2a}\sum_{t=1}^{n_{r_{ll}}-1}(d_{r_{ll}}[t]-\tilde{c}_{r_{ll}}[t-1])^2\right) + \right.$$

$$\left.\frac{1}{2}\exp\left(-\frac{1}{2a}\sum_{t=1}^{n_{r_l}-1}(d_{r_l}[t]-\tilde{c}_{r_l}[t-1])^2\right)\right) + \frac{1}{2}\exp\left(-\frac{1}{2a}\sum_{t=1}^{n_r-1}(d_r[t]-\tilde{c}_r[t-1])^2\right),$$

where the recursion is applied for all nodes $r_l, r_{ll}, \ldots$ until we reach the final node at depth K, i.e., r_{ll} in the last line of (15) for this example. Using Equation (15), $\tilde{P}_u(x^{n-1})$ can be compactly represented as sum of $K+1$ terms, collecting all terms that will not be affected by $x[n]$, i.e.,

$$\tilde{P}_u(x^{n-1}) = \sum_{k=0}^{K}\sigma_k[n-1]\exp\left(-\frac{1}{2a}\sum_{t=1}^{n_{\eta_k}-1}(d_{\eta_k}[t]-\tilde{c}_{\eta_k}[t-1])^2\right), \tag{16}$$

where, for this example, the dark nodes are labeled as $\eta_0 = r$, $\eta_1 = r_l$ and $\eta_2 = r_{ll}$. We will enumerate the dark nodes using the notation η_k, $k = 0, \ldots, K$. For each dark node η_k, $\sigma_k[n-1]$ contains products of node probabilities $\tilde{P}_\eta(x^n)$ that share the same parent nodes with η_k but will be unchanged by $x[n]$ (i.e., the sibling node of a dark node that does not include $x[n-1]$). As an example, consider the same tree of depth $K = 2$ in Figure 3 where we also included node probabilities. Then, it can be deduced from Figure 3 and Equation (15) that for each time $n-1$

$$\sigma_0[n-1] = \frac{1}{2},$$

$$\sigma_1[n-1] = \left(\frac{1}{2}\right)^2\tilde{P}_{r_u}(x^{n-1}) = \frac{1}{2}\tilde{P}_{r_u}(x^{n-1})\sigma_0[n-1],$$

$$\sigma_2[n-1] = \left(\frac{1}{2}\right)^3\tilde{P}_{r_u}(x^{n-1})\tilde{P}_{r_{lu}}(x^{n-1}) = \frac{1}{2}\tilde{P}_{r_{lu}}(x^{n-1})\sigma_1[n-1]$$

where for a tree, $K > 2$, in which $x[n-1]$ falls in the region for node r_{l^K} (a leaf), $\sigma_k[n-1] = \frac{1}{2}\tilde{P}_{r_{l^{k-1}u}}\sigma_{k-1}[n-1]$ (where we use short hand notation $l^k = ll^{k-1}$). Hence, at each time $n-1$, $\sigma_k[n-1]$ can be calculated recursively with only K updates. Clearly in the calculation of $\sigma_k[n-1]$, we use the nodes that will be unchanged by $x[n]$, i.e., $\tilde{P}_{r_u}(x^n) = \tilde{P}_{r_u}(x^{n-1})$, $\tilde{P}_{r_{lu}}(x^n) = \tilde{P}_{r_{lu}}(x^{n-1})$. Thus, to obtain

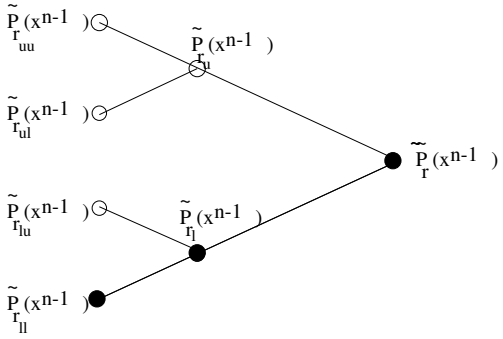


Fig. 3. Node probabilities at time $n-1$. Only dark nodes are to be updated, where each dark node uses a single sibling node to calculate $\sigma_k[n-1]$.

$\tilde{P}_u(x^n)$, we need to update only the exponential terms in Equation (15) or in Equation (16). Since also $d_r[n_r] = d_{r_l}[n_{r_l}] = d_{r_{ll}}[n_{r_{ll}}] = \ldots = x[n]$,

$$\begin{aligned}
&\tilde{P}_u(x^n) \\
&= \sum_{k=0}^{K} \sigma_k[n-1] \exp\left(-\frac{1}{2a} \sum_{t=1}^{n\eta_k - 1} (d_{\eta_k}[t] - \tilde{c}_{\eta_k}[t-1])^2\right) \exp\left(-\frac{1}{2a}(d_{\eta_k}[n\eta_k] - \tilde{c}_{\eta_k}[n\eta_k - 1])^2\right), \\
&= \sum_{k=0}^{K} \sigma_k[n-1] \exp\left(-\frac{1}{2a} \sum_{t=1}^{n\eta_k - 1} (d_{\eta_k}[t] - \tilde{c}_{\eta_k}[t-1])^2\right) \exp\left(-\frac{1}{2a}(x[n] - \tilde{c}_{\eta_k}[n\eta_k - 1])^2\right),
\end{aligned}$$

hence the sequential update for $\tilde{P}_u(x^n)$. A complete algorithmic description of this tree update with required storage and number of operations will be given in Section III-C.

Thus, $\tilde{P}_u(x[n]|x^{n-1})$ can be written

$$\begin{aligned}
\tilde{P}_u(x[n]|x^{n-1}) &= \frac{\tilde{P}_u(x^n)}{\tilde{P}_u(x^{n-1})} \\
&= \sum_{k=0}^{K} \mu_k[n-1] \exp\left(-\frac{1}{2a}(x[n] - \tilde{c}_{\eta_k}[n\eta_k - 1])^2\right),
\end{aligned}$$

where weights $\mu_k[n-1]$ are defined as

$$\mu_k[n-1] \stackrel{\triangle}{=} \frac{\sigma_k[n-1] \exp\left(-\frac{1}{2a} \sum_{t=1}^{n\eta_k - 1} (d_{\eta_k}[t] - \tilde{c}_{\eta_k}[t-1])^2\right)}{\tilde{P}_u(x^{n-1})}.$$

We are now ready to construct sequential prediction algorithms whose associated probability assignments asymptotically achieve $\tilde{P}_u(x^n)$ by upper bounding $\tilde{P}_u(x[n]|x^{n-1})$ at each time n. If we can find a prediction algorithm such that

$$\exp\left\{-\frac{1}{2a}(x[n] - \tilde{x}_c[n])^2\right\} \geq \tilde{P}_u(x[n]|x^{n-1}), \tag{17}$$

then we have achieved the desired result, that is, we will have a sequential prediction algorithm whose prediction error is asymptotically as small as that of the best predictor in the competition class. We will now introduce two different approaches to finding an $\tilde{x}_c[n]$ satisfying Equation (17). The first method is based on a concavity argument and results in an algorithm that can be constructed using a simple linear mixture. Although this approach results in a looser upper bound, it may be more suitable for adaptive filtering applications, given the reduced computational complexity. The second approach is based on the Aggregating Algorithm (AA) of [4] and requires a search, taking substantially greater, yet still polynomial time to construct each prediction. This second approach results the tighter upper bound introduced in Theorem 3. Both approaches use the same context tree and probabilities, and only differ in the last stage to form the final output.

Observe that $\tilde{P}_u(x[n]|x^{n-1})$ can be written

$$\tilde{P}_u(x[n]|x^{n-1}) = \sum_{k=0}^{K} \mu_k[n-1] f_n(\tilde{c}_{\eta_k}[n\eta_k - 1]), \tag{18}$$

where $f_t(.)$ is defined as

$$f_t(z) \stackrel{\triangle}{=} \exp\left(-\frac{(x[t]-z)^2}{2a}\right). \tag{19}$$

Since

$$\sum_{k=0}^{K} \mu_k[n-1] = 1,$$

$\tilde{P}_u(x[n]|x^{n-1})$ is sum of a function evaluated at a convex combination of values.

In the first method, if the function $f_t(.)$ is concave and $\sum_{i=1}^{n} \theta_i = 1$, then

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

by Jensen's inequality. The function defined in Equation (19) will be concave for values of z_i such that $(x[n]-z_i)^2 < a$. This corresponds to $-\sqrt{a} \leq (x[n]-\tilde{c}[n]) \leq \sqrt{a}$, where $\tilde{c}[n]$ is any prediction in Equation (18). Since the signal $|x[n]| \leq A_x$, then the prediction values in Equation (18) can be chosen such that $|\tilde{c}[n]| \leq A_x$. If the predicted values are outside this range, then the prediction error can only decrease by clipping. Therefore, by Jensen's inequality, whenever $a \geq 4A_x^2$ the function $f_t(.)$ will be concave at all points of the prediction and

$$\tilde{P}_u(x[n]|x^{n-1}) \leq \exp\left\{-\frac{1}{2a}\left(x[n] - \sum_{k=0}^{K} \mu_k[n-1]\tilde{c}_{\eta_k}[n-1]\right)^2\right\}$$

which gives the universal predictor as

$$\tilde{x}_c[n] = \sum_{k=0}^{K} \mu_k[n-1]\tilde{c}_{\eta_k}[n\eta_k - 1], \tag{20}$$

where η_k are the nodes such that $x[n-1] \in R_{\eta_k}$, i.e., dark nodes. By using Equation (14) we conclude that

$$\begin{aligned}
&\sum_{t=1}^{n}(x[t]-\tilde{x}_c[t])^2 \qquad (21)\\
&\leq 2aC(\mathcal{P}_i)\ln(2)+\inf_{c_{i,j}\in\mathcal{R}}\left\{\sum_{t=1}^{n}\left(x[t]-c_{i,s_i[t-1]}\right)^2+\delta\|\vec{c}_i\|^2\right\}+J_iA_x^2\ln(n/J_i)+O(1)\\
&\leq 8A_x^2C(\mathcal{P}_i)\ln(2)+\inf_{c_{i,j}\in\mathcal{R}}\left\{\sum_{t=1}^{n}\left(x[t]-c_{i,s_i[t-1]}\right)^2+\delta\|\vec{c}_i\|^2\right\}+J_iA_x^2\ln(n/J_i)+O(1).
\end{aligned}$$

For the second method, since $P_u(x[n] \mid x^{n-1})$ in Equation (18) is the sum of certain exponentials evaluated at a convex combination values, then for values of $a \geq A_x^2$ there exists an interval of $\tilde{x}_c[n]$ that satisfies Equation (17) and a value in this interval can be found in polynomial time [4]. Using this value of a yields an upper bound with one fourth the regret per node of that in Equation (21). Hence, using the AA of [4] in the final stage, instead of the convex combination, results in the following regret

$$\begin{aligned}
&\sum_{t=1}^{n}(x[t]-\tilde{x}_c[t])^2 \leq\\
&2A_x^2C(\mathcal{P}_i)\ln(2)+\inf_{c_{i,j}\in\mathcal{R}}\left\{\sum_{t=1}^{n}\left(x[t]-c_{i,s_i[t-1]}\right)^2+\delta\|\vec{c}_i\|^2\right\}+J_iA_x^2\ln(n/J_i)+O(1)
\end{aligned}$$

This concludes Proof of Theorem 2 for piecewise constant models. The proof of Theorem 2 for general affine models follows along similar lines. For construction of the universal algorithm, $\tilde{x}_w[n]$, we need only replace the prediction algorithm in Equation (11) with [7]

$$\tilde{c}_\eta[n] \stackrel{\triangle}{=} \tilde{\vec{w}}_\eta^T[n-1]\vec{y}[n], \qquad (22)$$

with

$$\tilde{\vec{w}}_\eta[n] = \left((Q_{\vec{y}_\eta\vec{y}_\eta}^{n\eta+1}+\delta I)^{-1}Q_{x_\eta\vec{y}_\eta}^{n\eta}\right)$$

where $\vec{y}[n]=[x[n-1]1]^T$, $Q_{\vec{y}_\eta\vec{y}_\eta}^{n\eta}=\sum_{t=1}^{n\eta}\vec{y}[t_\eta^{n\eta}]\vec{y}^T[t_\eta^{n\eta}]$, $Q_{x_\eta\vec{y}_\eta}^{n\eta}=\sum_{t=1}^{n\eta}x[t_\eta^{n\eta}]\vec{y}[t_\eta^{n\eta}]$, $\delta>0$ and I is an appropriate sized identity matrix. Here $x_\eta[t]$ and $\vec{y}_\eta[t]$ are the samples that belong to node η. By this replacement the universal algorithm is given by

$$\tilde{x}_w[n]=\sum_{k=0}^{K}\mu_k[n-1]\vec{w}_{\eta_k}^T[n-1]\vec{y}[n],$$

where η_k are the nodes such that $x[n-1] \in R_{\eta_k}$. This completes Proof of Theorem 2. ■

B. Outline of Proof of Theorem 3

Proof of Theorem 3 follows directly the proof of Theorem 2. We first update the definition of Equation (10) as,

$$\tilde{P}(x^n|\mathcal{P}_i) \stackrel{\triangle}{=} \exp\left(-\frac{1}{2a}\sum_{t=1}^{n}(x[t]-\hat{x}_{R_{i,s_i[t-1]}}[t])^2\right),$$

where $\hat{x}_{R_{i,j}}[t] = \hat{x}\eta[t]$ when $R_{i,j}$ is the region represented by the node η. The weighted probability of each node in Equation (12) is now defined as, $\tilde{P}_\eta(x^n) = \exp\left(-(1/2a)\sum_{t=1}^{n\eta}(d_\eta[t]-\hat{x}\eta[t])^2\right)$. Using the same recursion used in Equation (13), we again conclude $\tilde{P}_u(x^n) = \tilde{P}_r(x^n) = \sum_{\mathcal{P}_i} 2^{-C(\mathcal{P}_i)}\tilde{P}(x^n|\mathcal{P}_i)$ where r is the root node. After this point, we follow the Proof of Theorem 2 which concludes the outline of the proof of Theorem 3. ■

C. Algorithmic description

In this section we give a description of the final context tree prediction algorithm. A complete description is given in Figure 4.

For this implementation, given a context-tree of depth K, we will have $2^{K+1}-1$ nodes. Each node, indexed, $\eta = 1,\ldots,2^{K+1}-1$, has a corresponding predictor $C_\eta[n-1] = \tilde{\vec{w}}_\eta^T[n-1]\vec{y}[n]$ and two node variables, the total assigned probability of the node η

$$P_\eta[n-1] \stackrel{\triangle}{=} \tilde{P}_\eta(x^{n-1})$$

and the prediction performance of the node η

$$E_\eta[n-1] \stackrel{\triangle}{=} \exp\left(-\frac{1}{2a}\sum_{t=1}^{n\eta-1}(d_\eta[t]-\tilde{\vec{w}}_\eta^T[n-1]\vec{y}[n])^2\right).$$

Hence, for a full tree of depth K, we need to store a total of $3(2^{K+1}-1)$ variables. At each time $n-1$, only $K+1$ of these predictors or variables will be used or updated.

At each time $n-1$, we first determine the dark nodes, i.e., the nodes η_k such that $x[n-1] \in R_{\eta_k}$. For these nodes we calculate $\sigma_k[n-1]$ which are in turn to be used to calculate $\mu_k[n-1]$ and final output, after $O(K)$ operations. Here, each $\sigma_k[n-1]$ is recursively generated by the product of the probability of the corresponding sibling nodes $\tilde{P}_s(x^{n-1})$ and $\sigma_{k-1}[n-1]$. For the update, only the variables and the predictors of the selected nodes ($K+1$ of them) are updated using the new sample value $x[n]$. Hence, we efficiently combine N_K predictors only using $K+1$ predictions and $O(K+1)$ operations per prediction.

IV. 2-Dimensional Prediction with Context Tree Algorithm

For a 2-dimensional predictor, the predictions in each region can be given as a function of $x[n-1]$ and $x[n-2]$. The past observation space $[-A_x, A_x]^2$ is now divided into disjoint regions (areas) by the

Variables:

$\eta = 1, \ldots, 2^{K+1} - 1$:

$P_\eta[n-1] \stackrel{\triangle}{=} \tilde{P}_\eta(x^{n-1})$: Total node probability.

$E_\eta[n-1] \stackrel{\triangle}{=} \exp\left(-\frac{1}{2a}\sum_{t=1}^{n\eta^{-1}}(d_\eta[t] - \tilde{\vec{w}}_\eta^T[t-1]\vec{y}[n])^2\right)$: Prediction performance of node η.

$C_\eta[n-1] \stackrel{\triangle}{=} \tilde{\vec{w}}_\eta^T[t-1]\vec{y}[n]$: Prediction of node η for $x[n]$.

$\delta, \delta_1, \delta_2$: small, positive real constants.

A : Upper bound for the absolute value of the underlying process $|x[n]| < A$.

$\vec{d}[k]$: the kth component of vector $\vec{d}$.

Initialization:

For $\eta = 1, \ldots, 2^{K+1} - 1$: $P_\eta[0] = \delta_1^{-1}$, $E_\eta[0] = \delta_2^{-1}$, $C_\eta[0] = 0$.

For $k = 1, \ldots, K+1$: $\mu_k[0] = 0$ (initial weights of the universal predictor.), $\sigma_k[0] = 0$

Algorithm:

For $n = 1, \ldots, N$,

$\vec{d} = [\ ]$ (vector containing indices of dark nodes)

For $\eta = 1, \ldots, 2^{K+1} - 1$, (find dark nodes in $O(K)$ computations)

if $x[n-1] \in R_\eta$,

$\vec{d} = [\vec{d}; \eta]$

$\sigma_0[n-1] = \frac{1}{2}$ (find weight for each node)

For $\eta = \vec{d}[2], \ldots, \vec{d}[K+1]$,

$\sigma_k[n-1] = \frac{1}{2}P_s[n-1]\sigma_{k-1}[n-1]$ (where $R_{\vec{d}[k]} \bigcup R_s = R_{\vec{d}[k-1]}$

i.e., s is the sibling node of $\vec{d}[k]$)

$\mu_k[n-1] = \frac{\sigma_k[n-1]E_{\vec{d}[k]}[n-1]}{P_{\vec{d}[1]}[n-1]}$

$\tilde{x}_c[n] = \sum_{k=0}^{K}\mu_k[n-1]C_{\vec{d}[k]}[n-1]$ (prediction in $O(K)$ operations)

For $k = K+1, \ldots, 1$, (update node probabilities in $O(K)$ operations)

$E_{\vec{d}[k]}[n] = E_{\vec{d}[k]}[n-1]\exp\left(-\frac{1}{2a}(x[n] - C_{\vec{d}[k]}[n-1])^2\right)$

if $k = K+1$, $P_{\vec{d}[k]}[n] = P_{\vec{d}[k]}[n]$ (leaf node).

elseif $k \neq K+1$, $P_{\vec{d}[k]}[n] = \frac{1}{2}P_{\vec{d}[k]_u}[n-1]P_{\vec{d}[k]_l}[n-1] + \frac{1}{2}E_{\vec{d}[k]}[n]$.

$C_{\vec{d}[k]}[n] = \tilde{\vec{w}}_{\vec{d}[k]}^T[n]\vec{y}[n+1]$

Fig. 4. Complete algorithmic description of the context tree algorithm.

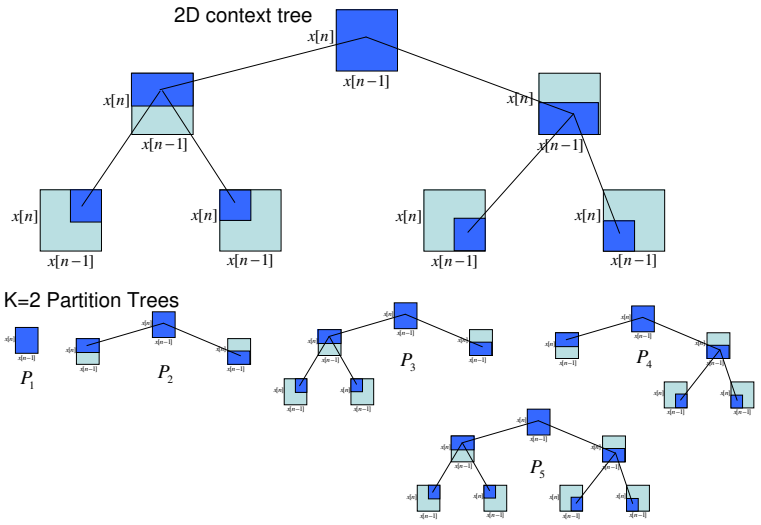


Fig. 5. Multi-dimensional extension: A 2-dimensional prediction using a context tree with $K = 2$. Each leaf in the context tree corresponds to a different quadrant in the real space $[-A_x, A_x]^2$, which is represented as a dark region in the figure. In the same figure, we also present the 5 different partitions represented by the $K = 2$ context tree algorithm. For each partition, again, the darker regions represents a leaf or a node. For each partition, the union of all dark regions results in the space $[-A_x, A_x]^2$.

context tree, $\bigcup_{j=1}^{J} S_j = [-A_x, A_x]^2$, as seen in Figure 5. Each area S_j is assigned to a leaf in the context tree. On this figure, we present a partition of $[-A_x, A_x]^2$ by a $K = 2$ context tree into 4 different regions, i.e., each leaf of the tree corresponds to a quadrant. For each region, the prediction is given by $\hat{x}[t] = w_{1,j}x[n-1] + w_{2,j}x[n-2] + c_j$, $w_{1,j} \in \mathcal{R}$, $w_{2,j} \in \mathcal{R}$, $c_j \in \mathcal{R}$ when $(x[n-1], x[n-2]) \in S_j$. For $K = 2$, there exist 5 different partitions as seen in Figure 5. Each of these partitions can be selected by the competing algorithm which then selects the corresponding 2-dimensional predictors in each region. After the selection of the context tree and the assignment of each region to the corresponding leaf, the algorithm proceeds as a one dimensional context tree algorithm. For each new sample $x[n-1]$, we again find the nodes corresponding to $(x[n-1], x[n-2])$ on the tree which are labeled as dark nodes. Then, we accumulate the corresponding probabilities based on the performance of each node. Only the prediction equations need to be changed to second order linear predictions.

In this section, we use the context tree method to represent a partition of the $(x[n-1], x[n-2])$ regressor space. The same context tree can be generalized to represent any partition of an arbitrary multi-dimensional space. Furthermore, a context tree can be used to represent more general state information. Here, the state information is derived from the membership of samples. The state information can be derived from an arbitrary source provided that the state information has a tree structure, i.e., membership in inner nodes infer membership in the corresponding leaves.

The algorithm from Theorem 2 can be extended to include p^{th}-order partitioning of the p-dimensional regressor space by a straightforward generalization, yielding the following result:

Theorem 4: *Let x^n be an arbitrary bounded real-valued sequence, such that $|x[t]| < A_x$ for all t. Then we can construct a sequential predictor $\tilde{x}_w[t]$ with complexity linear in the depth of the context tree per prediction such that*

$$\sum_{t=1}^{n}(x[t]-\tilde{x}_w[t])^2 \leq \inf_{\mathcal{P}_i}\left(\inf_{\vec{w}_{i,j}\in\mathcal{R}^p, c_{i,j}\in\mathcal{R}}\left\{\sum_{t=1}^{n}\left(x[t]-\vec{w}_{i,s_i[t-1]}^T\vec{y}[t]-c_{i,j}\right)^2+\delta(\|\vec{w}_i\|^2)\right\}+ \right. \quad (23)$$
$$\left. 8A_x^2C(\mathcal{P}_i)\ln(2)+(p+1)J_iA_x^2\ln(n/J_i)\right)+O(1),$$

where $\delta > 0$, and for a given p-dimensional partition $\mathcal{P}_i$ of the regressor space, the sequential algorithm competes with the vector of piecewise affine p^{th}-order prediction vectors. A similar sequential predictor with polynomial complexity can also be constructed. The proof of Theorem 4 is a straightforward generalization of that for Theorem 2.

V. Lower Bounds: Known Regions

To obtain lower bounds on the regret for any sequential predictor, we consider a set of J regions such that $x[t] \in R_j$ if $A_{x,j-1} < |x[t]| < A_{x,j}$, i.e., we consider a set of J regions which are concentric around the origin. Note that the upper bound in this case continues to be valid, since we do not make any assumption on the shape of the regions to obtain it, other than assuming that inside the jth region $|x[t]| < A_{x,j}$. For piecewise linear prediction, we have the following theorem.

Theorem 5: *Let x^n be an arbitrary bounded, real-valued sequence such that $|x[t]| < A_x$ for all t. Let $\hat{x}_q[t]$ be the predictions from any sequential prediction algorithm. Then*

$$\inf_{q\in\mathcal{Q}}\sup_{x^n}\frac{1}{n}\left\{l_n(x,\hat{x}_q)-\inf_{\vec{w}\in R^J}l_n(x,\hat{x}_{\vec{w}})\right\}\geq\frac{1}{n}\sum_{j=1}^{J}\frac{2C_j}{2C_j+1}A_{x,j}^2\ln\left(1+\frac{n_j-2}{2C_j}\right), \quad (24)$$

where $\mathcal{Q}$ is the class of all sequential predictors, C_j are positive constants, $l_n(x,\hat{x}_{\vec{w}}) = \sum_{t=1}^{n}(x[t] - w_{s[t-1]}x[t-1])^2$ and $s[t-1]$ is the indicator variable for the underlying partition with concentric regions around origin.

Theorem 5 provides a lower bound for the loss of any sequential predictor. Note that this bound depends on the values of $A_{x,j}$ and n_j (i.e., the number of samples inside each region). Since the values of $A_{x,j}$ are fixed and the lower bound holds for all values of n_j, $\sum_{j=1}^{J} n_j = n$ and n_j integer, n_j can be chosen to maximize the lower bound with the hope of asymptotically matching the upper bound derived in Theorem 1. A more general, but weaker, lower bound, can be derived by maximizing only with respect to n_j as

$$\inf_{q\in\mathcal{Q}}\sup_{x^n}\left\{l_n(x,\hat{x}_q)-\inf_{\vec{c}\in R^J}l_n(x,\hat{x}_{\vec{c}})\right\}\geq(1-\epsilon)\frac{\sum_{j=1}^{J}A_{x,j}^2}{J}\ln(n/J)-G,$$

for all $\epsilon > 0$.

A. Proof of Theorem 5

We begin by noting that for any distribution on x^n

$$\inf_{q\in\mathcal{Q}} \sup_{x^n} \left\{ l_n(x,\hat{x}_q) - \inf_{\vec{c}\in R^J} l_n(x,\hat{x}_{\vec{c}}) \right\} \geq \inf_{q\in\mathcal{Q}} \mathrm{E}_{x^n} \left\{ l(x^n,\hat{x}_q^n) - \inf_{\vec{c}\in R^J} l_n(x,\hat{x}_{\vec{c}}) \right\}, \tag{25}$$

where $\mathrm{E}_{x^n}(\cdot)$ is the expectation taken with respect to the distribution on x^n. Hence, to obtain a lower bound on the total regret, we just need to lower bound the right term in Equation (25).

Consider the following way of generating the sequence $x_j^{n_j}$. Let θ_j be a random variable drawn from a beta distribution with parameters (C_j, C_j), such that

$$p(\theta_j) = \frac{\Gamma(2C_j)}{\Gamma(C_j)\Gamma(C_j)} \theta_j^{C_j-1}(1-\theta_j)^{C_j-1},$$

where $C_j > 0$ is a constant, and $\Gamma(\cdot)$ is the gamma function. The sequence $x_j^{n_j}$ generated has only two possible values: $-A_{x,j}, A_{x,j}$. Hence, the sequence x^n can take a total of $2J$ different values: $-A_{x,J}, \ldots, -A_{x,1}, A_{x,0}, A_{x,1}, \ldots, A_{x,J}$. We generate the sequence in such a way that it spends the first n_1 points in the first region, the next n_2 points in the second region, and so on, spending the last n_J points in the Jth region. Obviously, $\sum_{j=1}^J n_j = n$. Inside each region, the sequence is generated such that $x[t] = x[t-1]$ with probability θ_j and $x[t] = -x[t-1]$ with probability $(1-\theta_j)$. In the transitions between regions, we generate the sequence such that $x[t] = A_{x,j+1}$ with probability $1/2$ and $x[t] = -A_{x,j+1}$ with probability $1/2$. Thus, given θ_j, any sequence $x_j^{n_j}$ forms a two-state Markov chain with transition probability $(1-\theta_j)$. The corresponding two states of the jth Markov chain are $-A_{x,j}$ and $A_{x,j}$. Hence, we have J Markov chains with transitions between them at predefined instants, and a probabilistic transition mechanism which determines the initial state of the $(j+1)$-th chain.

Given this distribution, we can now compute a lower bound for (25). Due to the linearity of the expectation, the right hand side of (25) becomes

$$L(n) = \inf_{q\in\mathcal{Q}} \mathrm{E}\{l_n(x,\hat{x}_q)\} - \mathrm{E}\{\inf_{\vec{w}} l_n(x,\hat{x}_{\vec{w}})\}, \tag{26}$$

where we drop the explicit dependence on x^n of the expectations to simplify notation. After this point, the proof of Theorem 5 directly follows from Theorem 2 of [7], where we apply the lower bound derived in Theorem 2 of [7] for each region separately. ■

VI. Simulations

In this section, we illustrate the performance of context tree algorithm with several examples. The first set of experiments involve prediction of a signal generated by a piecewise linear model by the following

equation,

$$x[t] = 0.1 * x[t-1] + 0.7 * x[t-2] + w[t], \text{ if } x[t-1] > 0 \text{ and } x[t-2] > 0 \quad (27)$$

$$x[t] = 0.1 * x[t-1] - 0.7 * x[t-2] + w[t], \text{ if } x[t-1] > 0 \text{ and } x[t-2] < 0$$

$$x[t] = 0.25 * x[t-1] + 0.1 * x[t-2] + w[t], \text{ if } x[t-1] < 0 \text{ and } x[t-2] > 0$$

$$x[t] = 0.9 * x[t-1] - 0.1 * x[t-2] + w[t], \text{ if } x[t-1] < 0 \text{ and } x[t-2] < 0$$

where $w[t]$ is a sample function from a stationary white Gaussian process of variance 1. Since the main results of this paper are on prediction of individual sequences, Figure 6a shows the normalized accumulated prediction error of our algorithms for a sample function of the process in Equation (27). Here, we use a 2-dimensional binary context-tree introduced in Section IV where $K = 4$ with second order linear predictors in each node. In the figures, we plot normalized accumulated prediction error for the context-tree algorithm, the sequential piecewise affine predictor that is tuned to the underlying partition in Equation (27) and the sequential algorithm corresponding to the finest partition. The underlying partition in Equation (27) corresponds to one of the partitions represented by the context-tree. The context-tree algorithm appears particularly useful for short data records. As expected, the performance of the finest partition suffers when data length is small, due to over-fitting. The context tree algorithm also outperforms the sequential predictor that is tuned to the underlying partition. Since the context tree algorithm adaptively combines predictors (for each different partition) based on their performance, it is able to favor the coarser models with a small number of parameters during the initial phase of the algorithm. This avoids the over-fitting problems faced by the sequential algorithms using the finest partition or the exact partition in Equation (27). As the data length increases, all three algorithms converge to the same minimum error rate. This makes the context-tree algorithm attractive for adaptive processing in time-varying environments for which a windowed version of the most recent data is typically used. Such applications require that algorithms continually operate in the short effective data length regime.

In Figure 6b, similar results to those in Figure 6a are presented and averaged over 100 different sample functions from Equation (27). The ensemble average performances and converge rates of each algorithm are similar to those for a single sample function.

In Figures 6a and 6b, the superior performance of the context-tree algorithm is shown with respect to the sequential algorithms corresponding to best partition and the true partition. As the data record increases, the context-tree algorithm also attains the performance of the best batch algorithm. Although the other sequential linear predictors will also asymptotically achieve their corresponding batch performance with different rates, the rate at which the context tree algorithm achieves the best batch performance and the performance of the best sequential algorithm is upper bounded by Theorem 2. These rates are at most

$O(C(\mathcal{P}_i)/n) + O(\ln(n)/n)$ and $O(C(\mathcal{P}_i)/n)$, respectively.

We next compare the performance of the context tree algorithm to a sequential algorithm using a recursive least squares (RLS) predictor with quadratic kernels. This set of experiments involve prediction of a signal generated by the following nonlinear equation,

$$x[t] = 0.1 * x[t-1] - 0.5 * (\cos(3 * x[t-1])) + 0.4 * \sin(x[t-2]) + 0.1 * x[t-2] + w[t], \quad (28)$$

where $w[t]$ is a sample function from a stationary white Gaussian process with unit variance. The RLS algorithm with quadratic kernels is given by

$$\hat{x}[t] = ax[t-1] + bx[t-2] + c(x[t-1])^2 + d(x[t-2])^2 + ex[t-1]x[t-2] \quad (29)$$

where each five parameters a, b, c, d, e are estimated using the RLS algorithm. Since the lattice implementation of the RLS algorithm would have complexity linear in the filter length per prediction, we compare it with a 1D context-tree algorithm which has $K = 4$ and linear predictors in each node, i.e.,$\hat{x} = wx[t-1]$ without the constant term. In Figure 7, we plot the normalized accumulated prediction error for the context-tree algorithm and the RLS algorithm with quadratic kernels for 100 trials. Again, the context-tree algorithm appears particularly useful for short data samples. The performance of the RLS algorithm attains the performance of the context tree algorithm as data lengths grows.

As the last example, we illustrate the performance of the context tree algorithm for a zero-mean sequence generated by removing the mean from a sample function of the Henon map, a chaotic process given by

$$x[n] = 1 - \alpha(x[n-1])^2 + \beta x[n-2] \quad (30)$$

and known to exhibit chaotic behavior for the values of $\alpha = 1.4$ and $\beta = 0.3$. The chaotic behavior of $x[n]$ can be seen in Figure 8, where we plot $x[n]$ given n. Although, $x[n]$ is chaotic, it is perfectly predictable, via Equation (30) given two prior samples. In Figure 9, we plot the normalized total square error (MSE) of several context tree prediction algorithms with different depths $K = 1, 2, 3, \ldots 10$, i.e., $\sum_{t=1}^{n}(x[t] - \hat{x}[t])^2$. Each context tree algorithm uses an affine predictor $\hat{x}[n] = w_j x[n-1] + c_j$ for prediction. We also plot the MSE of a linear predictor which uses the recursive least squares (RLS) algorithm. The order of the RLS predictor is 10. The context tree algorithms have superior performance with respect to the linear RLS predictor. The context tree algorithms are able to model the nonlinear term, $x[n-1]^2$, in the Henon map while the RLS predictor tries to approximate the nonlinearity with linear terms. The performance of the context tree algorithms improve as we increase the depth of the tree K. Although, the modeling power of the algorithms increase with the increased depth, the performance of the algorithms eventually saturate since the Henon map contains a second order term.

We then construct 2-dimensional context tree algorithms as in Section IV where we again try to predict the same Henon Map. In Figure 10, we plot the MSE of 1-dimensional and 2-dimensional context tree algorithms where each algorithm has depth $K = 8$. We plot context tree algorithms using constant predictors, $\hat{x}[n] = c_j$ and affine predictors $\hat{x}[n] = w_j x[n-1] + c_j$ in one dimension and constant predictors, $\hat{x}[n] = c_j$ and affine predictors $\hat{x}[n] = w_{1,j} x[n-1] + w_{2,j} x[n-1] + c_j$ in two dimensions. Since, the Henon Map is perfectly predictable, the MSE of the second order linear (affine) context tree algorithm continuously decreases with K.

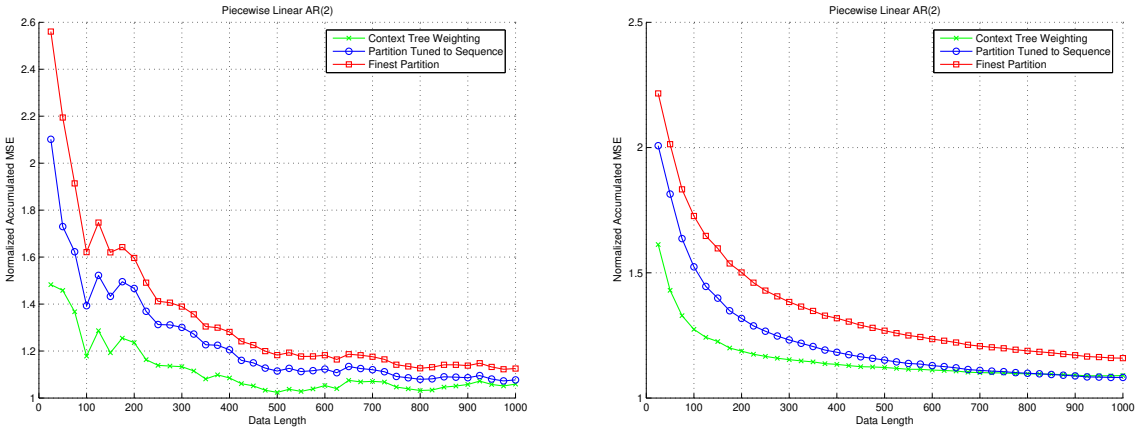
To observe the learning process of the context tree, we simulate the performance of a 2-dimensional context tree algorithm with depth $K = 10$ for the same Henon Map. In Figure 11, we plot the probability $P_\eta(x_1^n)$ assigned by the context tree algorithm to the predictor in each region of $[-A_x, A_x]^2$, for $n = 100, 500, 1000, 2000, 4000$ as a 2-D image. In the figure, darker regions correspond to smaller weights. Since the assigned probability of each region determines the contribution of its predictor to the final prediction, the larger the weight the greater the contribution of that region's prediction to the final output, from Equation (20). As n increases, the weight distribution of each region closely depicts the attractor of the Henon Map plotted in Figure 11a, i.e., the algorithm rapidly adapts to the underlying structure of the relation.

To further illustrate the operation of the context tree algorithm, Figure 12 depicts the probabilities assigned by the context tree algorithm to each level of the context tree for the same Henon map process. Here, we use a 2D context tree algorithm of depth-3 with affine predictors in each node. The probability assignments determines how much weight is given to the prediction of each partition in the final output by the context tree. On the figure, the first bar corresponds to the root probability. The second row (first level) has two bars for the two children, the third row(second level) has four bars for the four grandchildren and finally fourth row has 8 bars for 8 leaves. Figure 12 illustrates how the weights initially favor coarser partitions. As the data length increases the context tree algorithm shift its weights from coarser models to finer models.

From this representative set of simulations, we observe that the context tree algorithms provide considerable performance gains with respect to linear models (even with different effective window lengths) with similar computational complexity for variety of different applications. The unknown nonlinearity in the models are effectively resolved by the context tree approach.

VII. Conclusions

In this paper, we consider the problem of piecewise linear prediction from a competitive algorithm perspective. Using context trees and methods based on sequential probability assignment, we have shown a prediction algorithm whose total squared prediction error is within $O(\ln(n))$ of that of the best piecewise



(a) (b)

Fig. 6. (a) Prediction results for a sample function of the second-order piecewise linear process (27). The normalized accumulated sequential prediction error $l_n(x, \hat{x})/n$ for: a context-tree algorithm of depth-4 with second order predictors in each node; a sequential piecewise linear predictor that is tuned to the underlying partition as in (27); a sequential piecewise linear predictor with the finest partition on the context tree. (b) the same algorithms averaged over 100 trials.

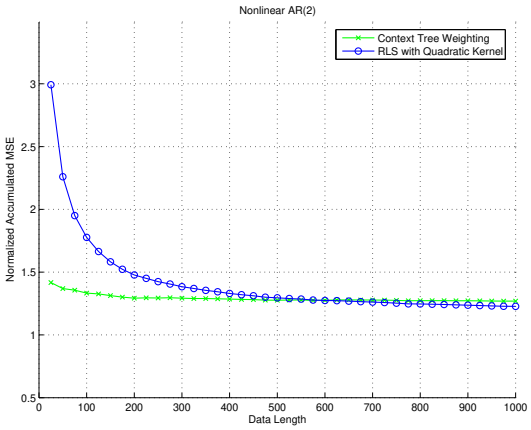


Fig. 7. Prediction results for a sample function of a nonlinear process given in Equation (28). The average normalized accumulated sequential prediction error for: a 1D binary context-tree algorithm with $K = 4$ using linear predictors at each node; a sequential algorithm using RLS with quadratic kernels as given in Equation (29).

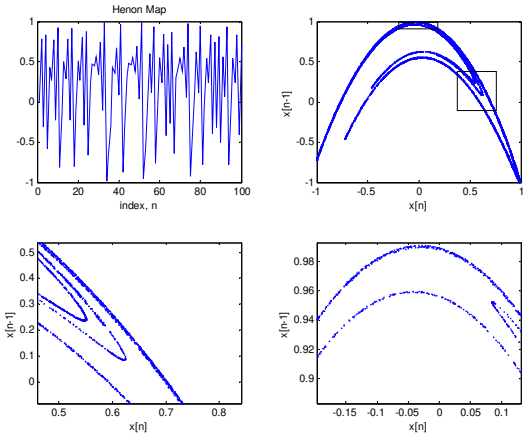


Fig. 8. Henon Map. $x[n] = 1 - 1.4x[n-1]^2 + 0.3x[n-2]$. The chaotic behavior of the Henon Map. Time evolution of $x[n]$ with respect to n. (upper left); $x[n]$ versus $x[n-1]$ (upper right); Zoomed version of the lower-rectangle (lower left); Zoomed version of the upper-rectangle (lower right).

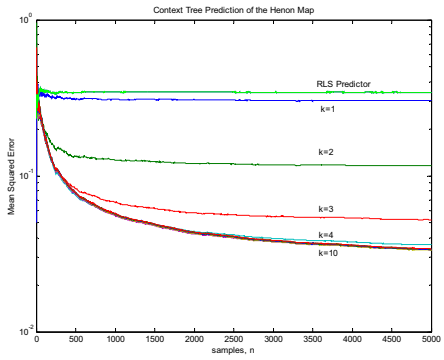


Fig. 9. Context tree prediction of the Henon Map. MSE performance of 1-dimensional context tree prediction algorithms with depths $K = 1, 2, 3, 4, \ldots, 10$ with uniform partition of the real line, using affine predictors. The Henon Map is given in Equation (30). Also, in the same figure, MSE performance of a linear predictor of order 10 using the RLS algorithm.

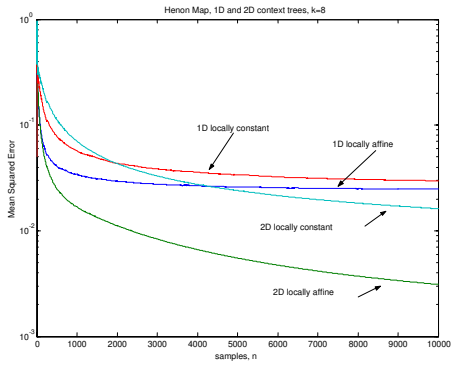


Fig. 10. Context tree prediction of the Henon Map. MSE performance of $K = 8$, 1-dimensional (scalar) and 2-dimensional context tree prediction algorithms with constant and linear (affine) predictors in each region. The Henon Map is given in Equation (30). Since the Henon map is perfectly predictable by 2-dimensional linear context tree algorithm, the MSE decreases continuously as n increases.

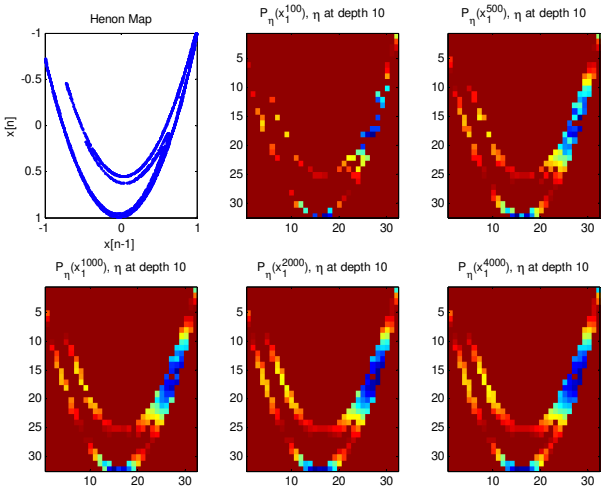


Fig. 11. Context tree prediction of the Henon Map. $P_\eta(x^n)$ is shown at depth 10 in the context tree for various times, $n = 100, 500, 1000, 2000,$ and 4000. The structure of the attractor becomes readily apparent as n increases.

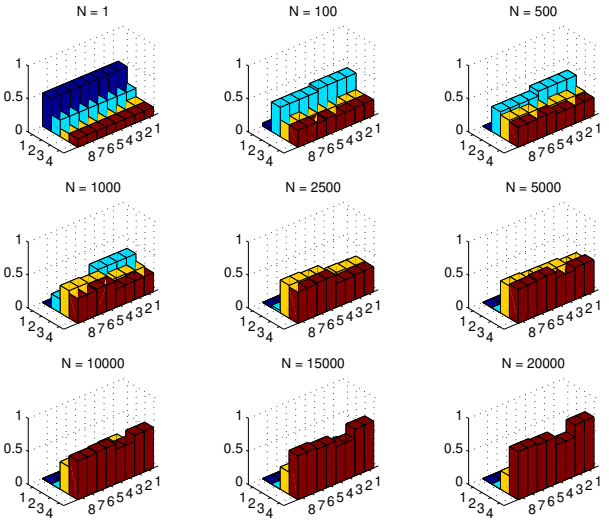


Fig. 12. 2D context tree prediction of the Henon Map. The weights assigned to the sequential predictors represented by the context-tree are shown. The first bar is the root probability. The second row has two bars for the two children, the third has four bars for the four grandchildren and finally fourth has 8 bars for 8 leaves. The heights are the node probabilities. corresponding to root node, first-level, second-level and third-level.

linear model tuned to the data in advance. We use a method similar to context tree weighting to compete well against a doubly exponential class of possible partitionings of the regressor space, for which we pay at most a "structural regret" proportional to the size of the best context tree. For each partition, we use a universal linear predictor to compete against the continuum of all possible affine models, for which we pay at most a "parameter regret" of $O(\ln(n))$. Upper and lower bounds on the regret are derived and scalar and vector prediction algorithms are detailed and demonstrated with examples. The resulting algorithms are efficient, with time complexity only linear in the depth of the context tree and perform well for a variety of data.

REFERENCES

[1] J. Makhoul, "Linear prediction: a tutorial review," *Proc. IEEE* 63:561-80, 1975.

[2] A. Singer, G. Wornell, and A. Oppenheim, "Nonlinear Autoregressive Modeling and Estimation in the Presence of Noise." *Digital Signal Processing*, vol. 4, no. 4, pp. 207-221, October 1994.

[3] P. Djuric, J. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, "Particle Filtering." *IEEE Magazine on Signal Processing,* vol. 20, no. 5, pp. 19-38, September 2003.

[4] V.Vovk, "Aggregating strategies" *COLT*, 1990, pp.371-383.

[5] V. Vovk, "Competitive on-line linear regression," *In: Advances in Neural Information Processing Systems (ed by M.I. Jordan, M.J. Kearns, and S.A. Solla)*, pp. 364–370, 1998.

[6] N. Cesa-Bianchi; P. M. Long; Warmuth, M.K., "Worst-case quadratic loss bounds for prediction using linear functions and gradient descent," *IEEE Transactions on Neural Networks*, Volume: 7 , Issue: 3 , May 1996 Pages:604 - 619

[7] A. C. Singer, S. S. Kozat, M. Feder, "Universal linear least squares prediction: upper and lower bounds," *IEEE Transactions on Information Theory*, vol. 48, no.8, pp. 2354-2362, Aug. 2002

[8] H. Tong. *Non-Linear Time Series: A Dynamical System Approach*. Oxford University Press, 1990.

[9] R.S. Tsay. "Testing and Modeling Threshold Autoregressive Processes." *Journal of the American Statistical Association,* vol. 84(405), pp. 231-240, 1989.
Reports

[10] T. Coulson, E.A. Catchpole, S.D. Albon, B.J.T. Morgan, J.M. Pemberton, T.H. Clutton-Brock, M.J. Crawley and B.T. Grenfell. "Age, Sex, Density, Winter Weather, and Population Crashes in Soay Sheep." *Science.* May, 2001, Vol. 292. no. 5521, pp. 1528 - 1531.

[11] M.P. Clements, and J. Smith. "A Monte Carlo Study of the Forecasting Performance of Empirical SETAR Models." *Journal of Applied Econometrics.* John Wiley & Sons, Ltd., vol. 14(2), pp.123-41, March-Apr 1999.

[12] Schoentgen, Jean. "Modelling the glottal pulse with a self-excited threshold auto-regressive model." In *EUROSPEECH'93*, pp. 107-110, 1993.

[13] A. C. Singer, M. Feder, "Universal linear prediction by model order weighting,"*IEEE Transactions on Signal Processing*, vol. 47, no. 10, October 1999.

[14] N. Merhav, M. Feder, "Universal schemes for sequential decision from individual data sequences," *IEEE Transactions on Information Theory,* , Volume: 39 , Issue: 4 , July 1993 Pages:1280 - 1292

[15] S.R. Kulkarni and S.E. Posner, "Universal Prediction of Nonlinear Systems," in *Proceedings of 34th Conference on Decision & Control*, pp. 4024–4029, Dec. 1995.

[16] M. Feder, N. Merhav, and M. Gutman, "Universal prediction of individual sequences", in *IEEE Transactions on Information Theory*, vol. 38, no. 4.,

[17] Willems, F.M.J.; Shtarkov, Y.M.; Tjalkens, T.J.; "The context-tree weighting method: basic properties," *I*EEE Transactions on Information Theory, Volume: 41 , Issue: 3 , May 1995 p.p. 653 - 664

[18] Sloane, N. J. A. Sequence A003095 (formerly M1544) in "The On-Line Encyclopedia of Integer Sequences."

[19] A. V. Aho, N. J. A. Sloane, "Some Doubly Exponential Sequences," *Fibonacci Quarterly*, vol. 11, pp. 429-437, 1970.

[20] D. P. Helmbold, R. E. Schapire, "Predicting nearly as well as the best pruning of a decision tree," *Machine Learning*, 27(1):51-68, 1997

[21] E. Takimoto, A. Maruoka and V. Vovk, "Predicting nearly as well as the best pruning of a decision tree through dyanamic programming scheme," *Theoretical Computer Science*, 261, 179-209, 2001

[22] E. Takimoto, M. K. Warmuth, "Predicting nearly as well as the best pruning of a planar decision graph" *Theoretical Computer Science*, 288, 217-235, 2002

[23] G. I. Shamir, N. Merhav, "Low-Complexity Sequential Lossless Coding for Piecewise-Stationary Memoryless Sources," *IEEE Transactions on Information Theory,* vol. 45, no. 5, pp.1498-1519, July 1999

[24] F. M. J. Willems, "Coding for a Binary Independent Piecewise-Identically-Distributed Source," *IEEE Transactions on Information Theory,* vol. 42, pp. 2210-2217, Nov. 1996

[25] D. S. Modha, E. Masry, "Universal, Nonlinear, Mean-Square Prediction of Markov Processes," in *Proceedings of IEEE On International Symposium on Information Theory*, p. 259, 1995.

[26] T. M. Cover, "Estimation by the nearest neighbor rule," in *IEEE Transactions on Information Theory*, vol. IT-14, pp. 50–55, Jan. 1968.

[27] Michel, O.J.J.; Hero, A.O., III; Badel, A.E., "Tree-structured nonlinear signal modeling and prediction," *IEEE Transactions on Signal Processing,* Volume: 47 , Issue: 11 , Nov. 1999 Pages: 3027 - 3041

[28] David Luengo, Suleyman S. Kozat, Andrew C. Singer, "Universal Piecewise Linear Least Squares Prediction: Upper and Lower Bounds," *I*nternational Symposium on Information Theory, Chicago, 2004

[29] R. E. Krichevsky and V. K. Trofi mov, "The Performance of Universal Encoding", *IEEE Transactions on Information Theory,* vol. 27, pp. 190-207, March 1981

[30] B. Ya. Ryabko, "Twice-universal coding," *Prob. Inf. Trans*, vol. 20, no. 3, pp. 173-7, 1984.