

Online Piecewise Linear Regression via Infinite Depth Context Trees

N. Denizcan Vanli, Muhammed O. Sayin, and Suleyman S. Kozat, *Senior Member, IEEE*

Abstract—We investigate the problem of sequential piecewise linear regression from a competitive framework. For an arbitrary and unknown data length n , we first introduce a method to partition the regressor space. Particularly, we present a recursive method that divides the regressor space into $O(n)$ disjoint regions (partitions) that can result in approximately 1.5^n different piecewise linear models on the regressor space. For each partition, we introduce a universal linear regressor whose performance is nearly as well as the best linear regressor whose parameters are set non-causally. We then use an infinite depth context tree to represent all piecewise linear models and introduce a universal algorithm to achieve the performance of the best piecewise linear model that can be selected in hindsight. In this sense, the introduced algorithm is twice-universal such that it sequentially achieves the performance of the best model that uses the optimal regression parameters. We then show that the performance of the introduced algorithm is asymptotically the same as the optimal twice differentiable function that is selected in hindsight. Our algorithm achieves this performance only with a computational complexity upper bounded by $O(n)$. Furthermore, we present a low complexity, namely $O(\log(n))$, implementation of the algorithm achieving the same upper bound under certain regularity conditions. We provide the explicit description of the algorithm as well as the upper bounds on the regret with respect to the best nonlinear and piecewise linear models, and demonstrate the performance of the algorithm through simulations.

Index Terms—Online, sequential, nonlinear, piecewise linear, regression, infinite depth context tree.

I. INTRODUCTION

NONLINEAR regression methods based on piecewise linear and locally linear approximations are extensively studied in order to capture the salient characteristics of a signal, where linear modeling yields unsatisfactory results [1]–[8]. Although nonlinear models are more powerful than the linear ones, their usage is generally limited due to the overfitting and convergence problems [1]–[3], [5], [7]. Therefore, in order to obtain a satisfactory performance while mitigating these issues, usually, tree based piecewise linear regressors are introduced instead of linear models [6]–[8].

In this brief, we consider the problem of sequential regression, where the aim is to estimate an unknown desired sequence $\{d[t]\}_{t \geq 1}$ by using a sequence of regressor vectors $\{x[t]\}_{t \geq 1}$. We refrain from any statistical assumptions on the unknown desired signal $\{d[t]\}_{t \geq 1}$ and the regressor vectors $\{x[t]\}_{t \geq 1}$, where the desired sequence and the regressor vectors are real valued and bounded, i.e., $d[t] \in \mathbb{R}$,

$x[t] \triangleq [x_1[t], \dots, x_p[t]]^T \in \mathbb{R}^p$ for an arbitrary integer p and $|d[t]|, |x_i[t]| < A < \infty$ for all t and $i = 1, \dots, p$. We call the regressors as “sequential” if in order to estimate the desired data at time t , i.e., $d[t]$, they only use the past information $d[1], \dots, d[t-1]$ and the observed regressor vectors,¹ $x[1], \dots, x[t]$. That is, say we have a sequential regressor $\hat{d}[t] = f(x[t])$, then the regressor function $f(\cdot)$ can be constructed using only d_1^{t-1} and x_1^t .

A simple and widely used regressor function is the linear regressor $f(x[t]) = w_t^T x_t$, where the weighting parameter w_t is updated at each time t according to an update rule, e.g., the recursive least squares (RLS) algorithm [9]. However, since the performance of a linear regressor may be unsatisfactory in many cases [1]–[8], instead of committing to a linear model, we partition the regressor space into disjoint regions and fit a separate linear model in each region. Furthermore, in order to efficiently manage the partitions defined on the regressor space, we use “context trees” [10], [11].

Although partitioning the regressor space to introduce non-linearity and using a tree structure to manage these partitions can be an efficient modeling method, the performance of the regressor is heavily affected by the construction of the tree [6]–[8]. Particularly, the “accurate” partitioning of the regressor space mainly defines the performance of the regressor. As an example, selection of the depth of the tree (i.e., the number of partitions) and the region boundaries of these partitions mainly define the performance of the regressor. While arbitrarily increasing the depth of the tree improves the modeling power of a regressor, such an increase usually results in overfitting [6]. Furthermore, although there exist methods that rely on held out data for such decisions, these methods usually do not have the theoretical justification or hard to implement in a sequential manner [7].

To overcome these issues, we do not directly commit to a fixed depth (and fixed power) tree, but introduce a method to construct a context tree [11], whose depth is adaptively incremented according to the unknown data length n . In this sense, the depth of the context tree goes to infinity as n , the data length, increases, hence we call such a tree as the “infinite depth context tree” [11]. Clearly, by defining such a partitioning method, we increase the number of disjoint regions on the regressor space as n increases. Therefore, the nonlinear modeling power of the regressor will increase sequentially as n increases, where the computational complexity of the introduced algorithm, in the worst-case scenario,

This work is supported in part by IBM Faculty Award, TUBITAK grants 112E161 and 113E517.

The authors are with the Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara 06800, Turkey (e-mail: {vanli, sayin, kozat}@ee.bilkent.edu.tr).

¹All vectors are column vectors and denoted by boldface lower case letters. Matrices are denoted by boldface upper case letters. For a vector x , x^T is the ordinary transpose. We denote $d_a^b \triangleq \{d[t]\}_{t=a}^b$.

is linear in the data length n , i.e., $O(n)$. We also present a low complexity implementation of the algorithm, that is logarithmic in n , i.e., $O(\log(n))$, and achieving asymptotically the same performance with the exact implementation under certain regularity conditions.

Hence, the main contributions of this brief are as follows. We introduce a sequential piecewise linear regression algorithm *i)* that provides a significantly improved modeling power by adaptively increasing the depth of the tree according to the arbitrary and unknown data length n , *ii)* that is highly efficient in terms of the computational complexity as well as the error performance, and *iii)* whose performance converges to *iii-a)* the performance of the optimal twice differentiable function that is selected in hindsight and *iii-b)* the best piecewise linear model defined on the infinite depth context tree, with guaranteed upper bounds without any statistical assumptions on the desired data. Hence, unlike the state of the art approaches whose performances usually depend on the initial construction of the tree, the introduced algorithm increases its nonlinear modeling power as the data length n increases, which results in a significantly superior performance. Furthermore, our algorithm achieves this performance only with a computational complexity $O(\log(n))$ under certain regularity conditions.

The organization of the paper is as follows. We first describe the sequential piecewise linear regression problem in detail in Section II. We then introduce an algorithm in Section III and prove that the performance of this algorithm is nearly as well as the best piecewise linear model that can be defined by the infinite depth context tree. Using this result, we also show that the introduced algorithm achieves the performance of the optimal twice differentiable function that is selected non-causally. In Section IV, we demonstrate the performance of our algorithm through simulations and then conclude the brief with several remarks in Section V.

II. PROBLEM DESCRIPTION

In the aforementioned framework, a piecewise linear model is constructed by dividing the regressor space into a union of disjoint regions, where in each region a linear model holds. As an example, suppose that the regressor space is parsed into K disjoint regions $\mathcal{R}_1, \dots, \mathcal{R}_K$ such that $\bigcup_{k=1}^K \mathcal{R}_k = [-A, A]^p$. Given such a model, say model m , at each time t , the sequential linear² regressor predicts $d[t]$ as $\hat{d}_m[t] = \mathbf{v}_{m,k}^T \mathbf{x}[t]$ when $\mathbf{x}[t] \in \mathcal{R}_k$, where $\mathbf{v}_{m,k}[t] \in \mathbb{R}^p$ for all $k = 1, \dots, K$.

However, by directly partitioning the regressor space as $\bigcup_{k=1}^K \mathcal{R}_k = [-A, A]^p$ before the processing starts and optimizing only the weighting parameters of the piecewise linear model, i.e., $\mathbf{v}_{m,k}[t]$, one significantly limits the performance of the regressor since we do not have any prior knowledge on the underlying desired signal. Therefore, instead of committing to a single piecewise linear model and performing optimization only over the regression parameters of this regressor, one can use a context tree to partition the regressor space, by which seeking to achieve the performance of the best partitioning over the whole doubly exponential number of different models represented by the context tree [12].

²Note that affine models can also be represented as linear models by appending a 1 to $\mathbf{x}[t]$, where the dimension of the regressor space increases by one.

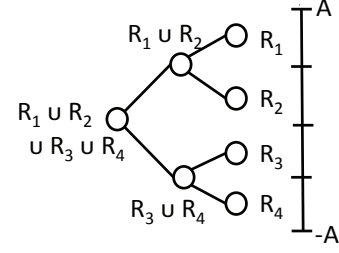


Fig. 1: The partitioning of a one dimensional regressor space, i.e., $[-A, A]$, using a depth-2 full context tree.

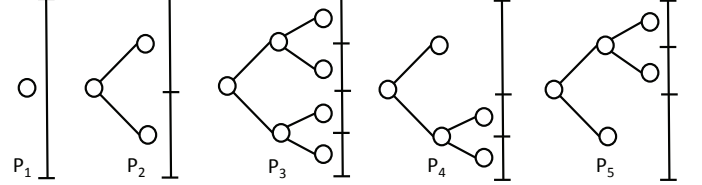


Fig. 2: All different piecewise linear models that can be obtained using a depth-2 full context tree, where the regressor space is one dimensional. These models are based on the partitioning shown in Fig. 1.

As an example, in Fig. 1, we partition the one dimensional regressor space using a depth-2 tree, where the regions $\mathcal{R}_1, \dots, \mathcal{R}_4$ correspond to the respective intervals on the real line and the internal nodes are constructed using these regions. In the generic case, for a depth- d full context tree, there exist 2^d leaf nodes and $2^d - 1$ internal nodes. Each node of the tree represents a portion of the regressor space such that the union of the regions represented by the leaf nodes is equal to the entire regressor space $[-A, A]^p$. Moreover, the region corresponding to each internal node is constructed by the union of the regions of its children. In this sense, we obtain $2^{d+1} - 1$ different regions on the regressor space and approximately 1.5^{2^d} different models that can be represented by depth- d tree [12]. We denote the set of all different piecewise linear models defined on a depth- d context tree as $\mathcal{M}_d$. As an example, we consider the same scenario as in Fig. 1, where we partition the one dimensional real space using a depth-2 context tree. Then, as shown in Fig. 2, a depth-2 tree defines $|\mathcal{M}_d| = 5$ different piecewise linear models, where each of these models is constructed using the nodes of the full depth context tree.

We emphasize that given a context tree of depth- d , the nonlinear modeling power of this tree is fixed and finite since there are only $2^{d+1} - 1$ different regions and approximately 1.5^{2^d} different nonlinear models defined on this tree. Instead of introducing such a limitation, we recursively increment the depth of the context tree as the data length increases. As previously mentioned, we call such a tree the “infinite depth context tree” [11], since the depth of the context tree goes to infinity as the data length n increases, hence in a certain sense, we can achieve an infinite nonlinear modeling power. That is, as n increases, the piecewise nonlinear models defined on the tree will converge to any unknown underlying nonlinear model under certain regularity conditions.

To this end, we try to minimize the following regret

$$\sum_{t=1}^n \left(d[t] - \hat{d}_s[t] \right)^2 - \inf_{m \in \mathcal{M}} \left\{ \inf_{\mathbf{v}_{m,k} \in \mathbb{R}^p} \sum_{k=1, \dots, K} \sum_{t=1}^n \left(d[t] - \hat{d}_b[t] \right)^2 \right\}, \quad (1)$$

for any n , where $\mathcal{M}$ denotes the set of all different piecewise linear models defined on the infinite depth context tree, $\mathbf{v}_{m,k}$ is the regression parameter of the k th partition of the m th piecewise linear model such that $\hat{d}_b[t] = \mathbf{v}_{m,k}^T \mathbf{x}[t]$ is the prediction of a batch regressor (when $\mathbf{x}[t] \in \mathcal{R}_k$), whose parameters can be set in hindsight after observing the entire data before processing starts. The term in (1) represents the difference in the performance of our algorithm and the optimal batch piecewise linear regressor embedded with the optimal regression parameters in hindsight. Therefore, an upper bound on (1) shows the convergence performance of the introduced algorithm.

III. NONLINEAR REGRESSION VIA INFINITE DEPTH CONTEXT TREES

In this section, we introduce a sequential piecewise linear regression algorithm that asymptotically achieves the performance of the best piecewise linear model defined on the infinite depth context tree and embedded with the optimum regression parameters. We provide the algorithmic details in the proof of Theorem 1.

Theorem 1: *Let $\{d[t]\}_{t \geq 1}$ and $\{\mathbf{x}[t]\}_{t \geq 1}$ be arbitrary, bounded, and real-valued sequences of data and regressor vectors, respectively. Then the algorithm $\hat{d}[t]$ given in Section III-A when applied to these data sequences yields*

$$\sum_{t=1}^n \left(d[t] - \hat{d}[t] \right)^2 - \inf_{m \in \mathcal{M}'} \left\{ \inf_{\mathbf{v}_{m,k} \in \mathbb{R}^p, k=1, \dots, K_m} \sum_{t=1}^n \left(d[t] - \hat{d}_b[t] \right)^2 \right\} \leq O(p \log^2(n)),$$

for any n , with a computational complexity upper bounded by $O(n)$, where $\mathcal{M}' \triangleq \{m \in \mathcal{M} : K_m \leq O(\log(n))\}$ and K_m represents the number of disjoint regions in model m .

This theorem implies that our algorithm given in Section III-A, asymptotically achieves the performance of the best piecewise linear model (having $O(\log(n))$ partitions), whose regression parameters are optimally set in hindsight, defined on the infinite depth context tree. Note that the number of different piecewise linear models defined on the infinite depth context tree can be in the order of 1.5^n [12]. This result indicates that as n increases, the performance of the introduced algorithm sequentially converges to the performance of more powerful piecewise linear regressors. Hence, as n increases, the difference in the performances of the introduced algorithm and the piecewise linear model that optimally partitions the regressor space will decrease. Such a powerful regression technique is achieved with a computational complexity upper bounded by $O(n)$, i.e., only linear in the data length.

Remark 1: Note that the algorithm in Section III-A achieves the performance of the best piecewise linear model having $O(\log(n))$ partitions with a regret of $O(p \log^2(n))$. For the models having $O(n)$ partitions, our algorithm still achieves a regret of $O(pn)$, where we emphasize that no other algorithm can also achieve a smaller regret than $O(pn)$ [6].

Before proving Theorem 1, we introduce another theorem, where we define the performance of our algorithm with respect to the class of twice differentiable functions.

Theorem 2: *Let $\{d[t]\}_{t \geq 1}$ and $\{\mathbf{x}[t]\}_{t \geq 1}$ be arbitrary, bounded, and real-valued sequences of data and regressor vectors, respectively. Let $\mathcal{F}$ be the class of all twice differentiable functions such that for any $f \in \mathcal{F}$, $\partial^2 f(\mathbf{x}) / \partial x_i \partial x_j \leq D < \infty$, $i, j = 1, \dots, p$ and we denote $\hat{d}_f[t] = f(\mathbf{x}[t])$. Then the algorithm $\hat{d}[t]$ given in Section III-A when applied to these data sequences yields*

$$\sum_{t=1}^n \left(d[t] - \hat{d}[t] \right)^2 - \inf_{f \in \mathcal{F}} \sum_{t=1}^n \left(d[t] - \hat{d}_f[t] \right)^2 \leq o(p^2 n),$$

for any n , with a computational complexity upper bounded by $O(n)$.

This theorem presents the nonlinear modeling power of the introduced algorithm and indicates that the introduced algorithm can sequentially achieve the asymptotically the same performance as the optimal twice differentiable function that is selected non-causally.

The proofs of Theorem 1 and 2 can be found in the following subsections, respectively.

A. Proof of Theorem 1 and Construction of the Algorithm

In order to prove Theorem 1, we first consider the parameter regret that results while learning the true regression parameters for a given piecewise linear model. We then introduce a method to partition the regressor space so that we obtain an infinite depth context tree. Finally, we consider the structural regret that results while learning the true partitioning of the regressor space for the introduced infinite depth context tree.

For the first part of the proof, consider that a piecewise linear model, say the m th model, having K_m disjoint regions $\mathcal{R}_1, \dots, \mathcal{R}_{K_m}$ such that $\bigcup_{k=1}^{K_m} \mathcal{R}_k = [-A, A]^p$ is given. Then, a piecewise linear regressor can be constructed using the universal linear predictor of [13] in each region as $\hat{d}_m[t] = \mathbf{v}_{m,k}^T \mathbf{x}[t]$, when $\mathbf{x}[t] \in \mathcal{R}_k$, with the regression parameters $\mathbf{v}_{m,k}[t] = (\mathbf{R}_k[t] + \delta \mathbf{I})^{-1} \mathbf{p}_k[t]$, where $\mathbf{I}$ represents the appropriate sized identity matrix, $\mathbf{R}_k[t] \triangleq \sum_{t' \leq t: \mathbf{x}[t'] \in \mathcal{R}_k} \mathbf{x}[t'] \mathbf{x}^T[t']$, and $\mathbf{p}_k[t] \triangleq \sum_{t' \leq t: \mathbf{x}[t'] \in \mathcal{R}_k} d[t'] \mathbf{x}[t']$. The upper bound on the performance of this regressor can be calculated following similar lines to [13] and it is obtained as follows

$$\sum_{t=1}^n \left(d[t] - \hat{d}_m[t] \right)^2 - \min_{\mathbf{v}_{m,k} \in \mathbb{R}^p, k=1, \dots, K_m} \left\{ \sum_{t=1}^n \left(d[t] - \hat{d}_b[t] \right)^2 + \delta \|\mathbf{v}_m\|^2 \right\} \leq A^2 K_m p \ln(n/K_m) + O(1). \quad (2)$$

This concludes the first part of the proof.

Before we introduce the partitioning method to generate the infinite depth context tree, we first introduce a labeling for the tree nodes following [10]. The root node is labeled with an empty binary string λ and assuming that a node has a label κ , where $\kappa = \nu_1 \dots \nu_l$ is a binary string of length l formed from letters $\nu_1, \dots, \nu_l$, we label its upper and lower children as $\kappa 1$ and $\kappa 0$, respectively. Here, we emphasize that a string can only take its letters from the binary alphabet, i.e., $\nu \in \{0, 1\}$, where 0 refers to the lower child, and 1 refers to the upper child of a node. We also introduce another concept, i.e., the definition of the prefix of a string. We say that a string $\kappa' = \nu'_1 \dots \nu'_{l'}$ is a prefix to string $\kappa = \nu_1 \dots \nu_l$ if $l' \leq l$ and $\nu'_i = \nu_i$ for all $i = 1, \dots, l'$, and the empty string λ is a prefix to all strings.

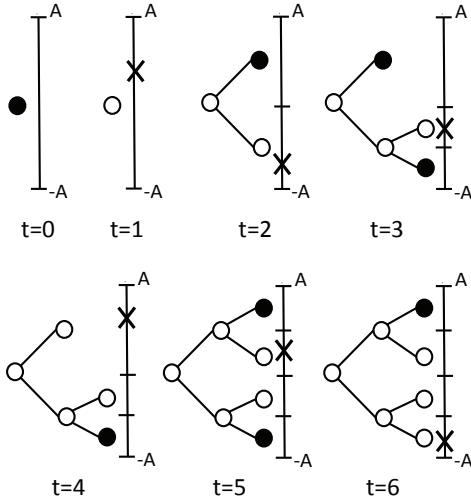


Fig. 3: A sample evolution of the infinite depth context tree, where the regressor space is one dimensional. The “x” marks on the regressor space represents the value of the regressor vector at that specific time instant. Light nodes are the ones having an index of 1, whereas the index of the dark nodes is 0.

Finally, we let $\mathcal{P}(\kappa)$ represent all prefixes to the string κ , i.e., $\mathcal{P}(\kappa) \triangleq \{\kappa_0, \dots, \kappa_l\}$, where $l \triangleq l(\kappa)$ is the length of the string κ , κ_i is the string with $l(\kappa_i) = i$, and $\kappa_0 = \lambda$ is the empty string, such that the first i letters of the string κ forms the string κ_i for $i = 0, \dots, l$.

Letting $\mathcal{L}$ denote the set of leaf nodes for a given context tree, we consider each leaf node of the tree $\kappa \in \mathcal{L}$, and define a specific index $\alpha_\kappa \in \{0, 1\}$ for these leaf nodes such that α_κ represents whether a regressor vector has fallen into $\mathcal{R}_\kappa$. That is, $\alpha_\kappa = 0$ represents that no regressor vector has fallen into region $\mathcal{R}_\kappa$, whereas $\alpha_\kappa = 1$ means that there was one. We also store the set of regressor vectors at each leaf node, which we denote by $\mathbf{x}_{\kappa,n} \triangleq \{\mathbf{x}[t], \forall t \in \{1, n\} : \mathbf{x}[t] \in \mathcal{R}_\kappa\}$.

We then present the algorithm to construct the infinite depth context tree as follows. At time $t = 0$, we begin with a single node, i.e., the root node λ , having index $\alpha_\lambda = 0$. Then, we recursively construct the context tree according to the following principle. For every time instant $t > 0$, we find the leaf node of the tree $\kappa \in \mathcal{L}$ such that $\mathbf{x}[t] \in \mathcal{R}_\kappa$. For this node if we have

- $\alpha_\kappa = 1$, then we generate two children nodes κ_0, κ_1 for this node by dividing the region $\mathcal{R}_\kappa$ into two disjoint regions $\mathcal{R}_{\kappa_0}, \mathcal{R}_{\kappa_1}$ using the plane $x_i = c$, where $i - 1 \equiv l(\kappa) \pmod{p}$ and c is the midpoint of the region $\mathcal{R}_\kappa$ along the i th dimension. Then, we divide the information stored in $\mathbf{x}_{\kappa,n}$ into $\mathbf{x}_{\kappa_0,n}, \mathbf{x}_{\kappa_1,n}$ and assign these sets to the nodes κ_0, κ_1 , respectively. Using this information, we calculate $\mathbf{v}_{m,\kappa_0}[t], \mathbf{v}_{m,\kappa_1}[t]$ and finally set $\alpha_{\kappa\nu} = 1$ for the node $\kappa\nu$, where $\nu \in \{0, 1\}$, such that $\mathbf{x}[t] \in \mathcal{R}_{\kappa\nu}$, and set $\alpha_{\kappa\nu^c} = 0$, where ν^c represents the complementary letter of ν in the binary alphabet $\{0, 1\}$.
- $\alpha_\kappa = 0$, then we only increment this number by 1 and perform the algorithmic updates without any modification on the context tree.

As an example, in Fig. 3, we consider that the regressor space is one dimensional and present a sample evolution of the tree, where in the figure, the nodes having an index of 0

are shown as dark nodes, whereas the others are light nodes, and the regressor vectors are marked with ‘x’ in the one dimensional regressor space. For instance at time $t = 2$, we have a depth-1 context tree, where we have two nodes 0 and 1 with corresponding regions $\mathcal{R}_0 = [-A, 0]$, $\mathcal{R}_1 = [0, A]$, and $\alpha_0 = 1, \alpha_1 = 0$. Then, at time $t = 3$, we observe a regressor vector $\mathbf{x}[3] \in \mathcal{R}_0$ and divide this region into two disjoint regions using $x_1 = -A/2$ line. We then find that in fact $\mathbf{x}[3] \in \mathcal{R}_{01}$, hence set $\alpha_{01} = 1$, whereas $\alpha_{00} = 0$.

Clearly, using this method, the number of light nodes on the tree is t at time t , therefore we may observe a context tree of depth n in the worst-case scenario, e.g., when $\mathbf{x}[t] = [A, \dots, A]^T$ for all t . Hence, the computational complexity of the algorithm is upper bounded by $O(n)$. This concludes the second part of the proof, i.e., the construction of the infinite depth context tree.

In the final part of the proof, we consider the structural regret of our algorithm. We first assign a weight based on the performance [10] for each leaf node $\kappa \in \mathcal{L}$ as follows

$$P_\kappa(n) \triangleq \exp \left\{ -\frac{1}{2a} \sum_{t \leq n : \mathbf{x}[t] \in \mathcal{R}_\kappa} \left(d[t] - \hat{d}_{m,k}[t] \right)^2 \right\},$$

where $\hat{d}_{m,k}[t]$ is constructed using the regressor introduced in [13] and discussed in the first part of the proof. Then, we define the probability of an inner node $\kappa \notin \mathcal{L}$ as follows

$$P_\kappa(n) \triangleq \frac{1}{2} P_{\kappa_0}(n) P_{\kappa_1}(n) + \frac{1}{2} \exp \left\{ -\frac{1}{2a} \sum_{t \leq n : \mathbf{x}[t] \in \mathcal{R}_\kappa} \left(d[t] - \hat{d}_{m,k}[t] \right)^2 \right\}.$$

Using this definitions, we then construct the weight of the root node λ as follows $P_\lambda(n) = \sum_{m \in \mathcal{M}} 2^{-B_m} P(n|m)$, where $P(n|m) \triangleq \exp \left\{ -\frac{1}{2a} \sum_{t=1}^n \left(d[t] - \hat{d}_m[t] \right)^2 \right\}$ represents the performance of a given partition $m \in \mathcal{M}$ over a data length of n , and B_m represents the number of bits required to represent the model m on the context tree using a universal code [14].

Hence, the performance of the root node satisfies $P_\lambda(n) \geq 2^{-B_m} P(n|m)$ for any $m \in \mathcal{M}$. That is,

$$-2a \ln(P_\lambda(n)) \leq \min_{m \in \mathcal{M}} \left\{ \sum_{t=1}^n \left(d[t] - \hat{d}_m[t] \right)^2 \right\} + 2a \ln(2) \log(n) + 4A^2 K_m \log(n), \quad (3)$$

where the last line follows when we maximize B_m with respect to $m \in \mathcal{M}$ and the regret term $4A^2 K_m \log(n)$ follows due to the adaptive construction of the infinite depth context tree.

Hence, we have obtained a weighting assignment achieving the performance of the optimal piecewise linear model. We next introduce a sequential algorithm achieving $P_\lambda(n)$. To this end, we first note that $P_\lambda(n) = \prod_{t=1}^n P_\lambda(t|t-1)$, where $P_\lambda(t|t-1) \triangleq P_\lambda(t)/P_\lambda(t-1)$. Now if we can demonstrate a sequential algorithm whose performance is greater than or equal to $P_\lambda(t|t-1)$ for all t , we can conclude the proof. That is, we present a sequential update from $P_\lambda(t-1)$ to $P_\lambda(t)$.

After the structural updates (which are already discussed in the second part of the proof) are completed, say at time t , we observe a regressor vector $\mathbf{x}[t] \in \mathcal{R}_\kappa$ for some $\kappa \in \mathcal{L}$. Then,

we can compactly denote the weight of the root node at time $t - 1$ as follows

$$P_\lambda(t-1) = \sum_{\kappa_i \in \mathcal{P}(\kappa)} \pi_{\kappa_i}[t-1] \times \exp \left\{ -\frac{1}{2a} \sum_{t' < t: \mathbf{x}[t'] \in \mathcal{R}_{\kappa_i}} \left(d[t'] - \hat{d}_{\kappa_i}[t'] \right)^2 \right\},$$

where $\hat{d}_\kappa[t]$ represents the output of the regressor for node κ , $\kappa_i \in \mathcal{P}(\kappa)$ is the string formed from the first i letters of $\kappa = \nu_1 \dots \nu_l$, and $\pi_{\kappa_i}[t]$ is recursively defined as follows

$$\pi_{\kappa_i}[t] \triangleq \begin{cases} \frac{1}{2} & , \text{ if } i = 0 \\ \frac{1}{2} P_{\kappa_{i-1}\nu_i^c}(t-1) \pi_{\kappa_{i-1}}[t] & , \text{ if } 1 \leq i \leq l-1 \\ P_{\kappa_{i-1}\nu_i^c}(t-1) \pi_{\kappa_{i-1}}[t] & , \text{ if } i = l \end{cases}$$

Since $\mathbf{x}[t] \in \mathcal{R}_\kappa$ for some $\kappa \in \mathcal{L}$, then after $d[t]$ is revealed, the weight of the root node at time t can be calculated as follows

$$P_\lambda(t) = \sum_{\kappa_i \in \mathcal{P}(\kappa)} \pi_{\kappa_i}[t-1] \exp \left\{ -\frac{1}{2a} \left(d[t] - \hat{d}_{\kappa_i}[t] \right)^2 \right\} \times \exp \left\{ -\frac{1}{2a} \sum_{t' < t: \mathbf{x}[t'] \in \mathcal{R}_{\kappa_i}} \left(d[t'] - \hat{d}_{\kappa_i}[t'] \right)^2 \right\},$$

which results in

$$P_\lambda(t|t-1) = \frac{P_\lambda(t)}{P_\lambda(t-1)} = \sum_{\kappa_i \in \mathcal{P}(\kappa)} \mu_{\kappa_i}[t-1] \exp \left\{ -\frac{1}{2a} \left(d[t] - \hat{d}_{\kappa_i}[t] \right)^2 \right\}, \quad (4)$$

where

$$\mu_{\kappa_i}[t-1] \triangleq \frac{\pi_{\kappa_i}[t-1] \exp \left\{ -\frac{1}{2a} \sum_{t' < t: \mathbf{x}[t'] \in \mathcal{R}_{\kappa_i}} \left(d[t'] - \hat{d}_{\kappa_i}[t'] \right)^2 \right\}}{P_\lambda(t-1)}.$$

We then focus on (4) and observe that we have $\sum_{\kappa_i \in \mathcal{P}(\kappa)} \mu_{\kappa_i}[t-1] = 1$, which means that if the second term in (4), i.e., $f(\hat{d}_{\kappa_i}[t]) \triangleq \exp \left\{ -\frac{1}{2a} \left(d[t] - \hat{d}_{\kappa_i}[t] \right)^2 \right\}$, is concave, then by Jensen's inequality, we can conclude that

$$\exp \left\{ -\frac{1}{2a} \left(d[t] - \sum_{\kappa_i \in \mathcal{P}(\kappa)} \mu_{\kappa_i}[t-1] \hat{d}_{\kappa_i}[t] \right)^2 \right\} \geq P_\lambda(t|t-1). \quad (5)$$

Since the function $f(\hat{d}_{\kappa_i}[t])$ is concave when $\left(d[t] - \hat{d}_{\kappa_i}[t] \right)^2 < a$, and we have $|d[t]| \leq A$, we have to set $a \geq 4A^2$. Hence, we obtain a sequential regressor in (5), whose performance is greater than or equal to the performance of the root node.

Finally, we define the final estimate of our algorithm $\hat{d}[t] \triangleq \sum_{\kappa_i \in \mathcal{P}(\kappa)} \mu_{\kappa_i}[t-1] \hat{d}_{\kappa_i}[t]$ and combining the results in (2), (3), and (5), we obtain

$$\begin{aligned} & \sum_{t=1}^n \left(d[t] - \hat{d}[t] \right)^2 \\ & \leq \min_{m \in \mathcal{M}} \left\{ \min_{\mathbf{v}_{m,k} \in \mathbb{R}^p} \left\{ \sum_{k=1, \dots, K_m}^n \left(d[t] - \hat{d}_b[t] \right)^2 + \delta \|\mathbf{v}_m\|^2 \right\} \right\} \\ & + \underbrace{A^2 K_m (p \ln(n/K_m) + 4 \log(n)) + 2a \ln(2) \log(n) + O(1)}_{\leq O(p \log^2(n))}, \end{aligned}$$

where the upper bound on the regret follows when $K_m = O(\log(n))$. This concludes the proof of Theorem 1 and construction of the algorithm. $\square$

Remark 2: Note that as mentioned in Remark 1 (and also can be observed in (2)), no algorithm can converge to the performance of the piecewise linear models having $O(n)$ disjoint regions. Therefore, we can limit the maximum depth of the tree by $O(\log(t))$ at each time t to achieve a low complexity implementation. With this limitation and according to the update rule of the tree, we can observe that while dividing a region into two disjoint regions, we may be forced to perform $O(n)$ computations due to the accumulated regressor vectors. However, since a regressor vector is processed by at most $O(\log(n))$ nodes for any n , the average computational complexity of the update rule of the tree remains $O(\log(n))$. Furthermore, the performance of this low complexity implementation will be asymptotically the same as the exact implementation provided that the regressor vectors are evenly distributed in the regressor space, i.e., they are not gathered around a very small neighborhood. This result follows when we multiply the tree construction regret in (3) by the total number of accumulated regressor vectors, whose order, according to the above condition, is upper bounded by $o(n)$.

B. Proof of Theorem 2

We begin our proof by emphasizing that the introduced algorithm converges to the best linear model in each region with a regret of $O(p \log^2(n))$ as already proven in Theorem 1. Therefore, using any other linear model yields a higher regret. Hence, say we define a suboptimal affine model by applying Taylor's theorem to a twice differentiable function $f \in \mathcal{F}$ about the midpoint of each region. Let $\hat{d}_s[t]$ denote the prediction of this suboptimal affine regressor. Then, we have

$$\sum_{t=1}^n \left(d[t] - \hat{d}[t] \right)^2 \leq \sum_{t=1}^n \left(d[t] - \hat{d}_s[t] \right)^2 + O(p \log^2(n)).$$

Now applying the mean value theorem with the Lagrange form of the remainder, we obtain

$$\begin{aligned} & \sum_{t=1}^n \left(d[t] - \hat{d}_s[t] \right)^2 - \sum_{t=1}^n \left(d[t] - \hat{d}_f[t] \right)^2 \\ & \leq 2A \sum_{t=1}^n \left\{ \sum_{i=1}^p \sum_{j=1}^p \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \Big|_{\mathbf{x}=\mathbf{b}} (x_i[t] - a_{\kappa,i})(x_j[t] - a_{\kappa,j}) \right\}, \end{aligned}$$

for some $m \in \mathcal{M}'$ and $\mathbf{b} \in \mathcal{R}_\kappa$, where $\mathbf{a}_\kappa = [a_{\kappa,1}, \dots, a_{\kappa,p}]^T$ is the midpoint of the region $\mathcal{R}_\kappa$. Maximizing this upper bound with respect to $\mathbf{x}$ we obtain

$$\begin{aligned} & \sum_{t=1}^n \left(d[t] - \hat{d}[t] \right)^2 - \sum_{t=1}^n \left(d[t] - \hat{d}_f[t] \right)^2 \\ & \leq 2ADp^2n \frac{A^2}{O(\log^{2/p}(n))} + O(p \log^2(n)) \\ & \leq o(p^2n). \end{aligned}$$

This concludes the proof of Theorem 2. $\square$

IV. SIMULATIONS

In this section, we illustrate the performance of our algorithm under well-known scenarios with respect to various

Algorithms Data Sets	IDCTW	IDCTW-lc	CTW	LR	VF	FNF
Lorenz Attractor	0.0354	0.0354	0.0360	0.0422	0.0371	0.0403
Mackey-Glass Sequence	0.0244	0.0244	0.0291	0.1031	0.0354	0.0373
Chua's Circuit	0.0266	0.0266	0.0316	0.0372	0.0372	0.0802
Duffing Map	0.0124	0.0124	0.0933	0.2200	0.2308	0.5977
Rössler Attractor	0.1468	0.1468	0.1798	0.1983	0.1765	0.1580
Tinkerbell Map	0.0834	0.0834	0.1068	0.1083	0.1034	0.0885

TABLE I: Time accumulated normalized errors of the proposed algorithms. Each dimension of the data sets is normalized between $[-1, 1]$.

methods. Results of the experiments are given in Table I, where we denote the infinite depth context tree weighting algorithm of Theorem 1 by “IDCTW”, its low complexity implementation in Remark 2 by “IDCTW-lc”, the context tree weighting algorithm of [6] having depth 2 by “CTW”, the linear regressor by “LR”, the second order Volterra filter by “VF” [9], and the second order Fourier nonlinear filter of [15] by “FNF”. Note that the combination weights of the LR, VF, and FNF are updated using the RLS algorithm.

The Lorenz attractor is defined by the following differential equations $\frac{dx}{dt} = \sigma(y-x)$, $\frac{dy}{dt} = x(\rho-z)-y$, and $\frac{dz}{dt} = xy-\beta z$, where we set $\rho = 28$, $\sigma = 10$, and $\beta = 8/3$.

The Mackey-Glass sequence is defined by the following differential equation $\frac{dx}{dt} = \frac{\beta x_\tau}{1+x_\tau^n} - \gamma x$, where x_τ denotes the value of x at time $t - \tau$ and we set $\beta = 2$, $\gamma = 1$, $\tau = 2$, and $n = 10$ with the initial condition $x = 0.5$ for $t < 0$. We generate the time series using the fourth order Runge-Kutta method.

The Chua’s circuit is generated according to the following differential equations $\frac{dx}{dt} = \alpha(y-x-f(x))$, $\frac{dy}{dt} = x-y+z$, and $\frac{dz}{dt} = -\beta y$, where $f(x) = m_1 x + 0.5(m_0 - m_1)(|x+1| - |x-1|)$, $\alpha = 15.6$, $\beta = 28$, $m_0 = -1.143$, and $m_1 = -0.714$ with initial conditions $[x, y, z] = [0.7, 0, 0]$ for $t < 0$.

The Duffing map is generated by the following discrete time equation $x_{t+1} = ax_t - x_t^3 - bx_{t-1}$, where we set $a = 2.75$ and $b = 0.2$ to produce the chaotic behavior.

The Rössler attractor is defined by the following differential equations $\frac{dx}{dt} = -y-z$, $\frac{dy}{dt} = x+ay$, and $\frac{dz}{dt} = b+z(x-c)$, where we set $a = b = 0.2$ and $c = 5.7$.

The Tinkerbell map is generated by the following discrete time equations $x_{t+1} = x_t^2 - y_t^2 + ax_t + by_t$ and $y_{t+1} = 2x_t y_t + cx_t + dy_t$, where we set $a = 0.9$, $b = -0.6013$, $c = 2$, and $d = 0.5$.

We emphasize that these values are selected to generate the well-known chaotic behavior of the corresponding attractors and maps, where the data lengths are 2500 for the Chua’s circuit, 5000 for the Lorenz attractor and the Mackey-Glass sequence, and 10000 for the Duffing map, the Rössler attractor and the Tinkerbell map to observe the convergence performances of the proposed algorithms.

As can be observed from Table I, the introduced algorithm significantly outperforms its competitors and achieves an average of 20% relative gain in the performance compared to its competitors. We emphasize that this improved performance is achieved by the incremental nonlinear modeling power of the introduced algorithm. Note that since the IDCTW algorithm achieves the performance of any twice-differentiable function, we expect that the error of the IDCTW algorithm converges to zero as n increases. Furthermore, we can observe that the performances of the IDCTW and the IDCTW-lc algorithms

are exactly the same, hence the IDCTW-lc algorithm can be used as an elegant alternative to the IDCTW algorithm.

V. CONCLUDING REMARKS

We study nonlinear regression of deterministic signals using an infinite depth context tree, where the regressor space is partitioned using a nested structure and independent regressors are assigned to each region. In this framework, we introduce a tree based algorithm that sequentially increases its nonlinear modeling power and achieves the performance of the optimal twice-differentiable function as well as the performance of the best piecewise linear model defined on the infinite depth context tree. Furthermore, this performance is achieved only with a computational complexity logarithmic in the data length n , i.e., $O(\log(n))$ (under regularity conditions). We illustrate the merits of the introduced algorithm over a series of well-known nonlinear problems in the literature.

REFERENCES

- [1] L. Devroye, T. Linder, and G. Lugosi, “Nonparametric estimation and classification using radial basis function nets and empirical risk minimization,” *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 475–487, 1996.
- [2] A. Krzyzak and T. Linder, “Radial basis function networks and complexity regularization in function learning,” *IEEE Transactions on Neural Networks*, vol. 9, no. 2, pp. 247–256, 1998.
- [3] I. G. Ali and Y.-T. Chen, “Design quality and robustness with neural networks,” *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1518–1527, 1999.
- [4] R. Gribonval, “From projection pursuit and CART to adaptive discriminant analysis?” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 522–532, 2005.
- [5] A. C. Singer, G. W. Wornell, and A. V. Oppenheim, “Nonlinear autoregressive modeling and estimation in the presence of noise,” *Digital Signal Processing*, vol. 4, no. 4, pp. 207–221, 1994.
- [6] S. S. Kozat, A. C. Singer, and G. C. Zeitler, “Universal piecewise linear prediction via context trees,” *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3730–3745, 2007.
- [7] S. Dasgupta and Y. Freund, “Random projection trees for vector quantization,” *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3229–3242, 2009.
- [8] Y. Yilmaz and S. S. Kozat, “Competitive randomized nonlinear prediction under additive noise,” *IEEE Signal Processing Letters*, vol. 17, no. 4, pp. 335–339, 2010.
- [9] A. H. Sayed, *Fundamentals of Adaptive Filtering*. NJ: John Wiley & Sons, 2003.
- [10] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, “The context-tree weighting method: basic properties,” *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 653–664, 1995.
- [11] F. M. J. Willems, “The context-tree weighting method: extensions,” *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 792–798, 1998.
- [12] A. V. Aho and N. J. A. Sloane, “Some doubly exponential sequences,” *Fibonacci Quarterly*, vol. 11, pp. 429–437, 1970.
- [13] A. C. Singer, S. S. Kozat, and M. Feder, “Universal linear least squares prediction: upper and lower bounds,” *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2354–2362, 2002.
- [14] F. M. J. Willems, “Coding for a binary independent piecewise-identically-distributed source,” *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 2210–2217, 1996.
- [15] A. Carini and G. L. Sicuranza, “Fourier nonlinear filters,” *Signal Processing*, vol. 94, no. 0, pp. 183 – 194, 2014.