TWICE-UNIVERSAL PIECEWISE LINEAR REGRESSION VIA INFINITE DEPTH CONTEXT TREES

N. Denizcan Vanli, Muhammed O. Sayin*, Tolga Göze†, and Suleyman S. Kozat**

* Department of Electrical and Electronics Engineering
Bilkent University, Bilkent, Ankara 06800, Turkey
E-mail: {vanli, sayin, kozat}@ee.bilkent.edu.tr.

† Alcatel-Lucent, Istanbul, Turkey
Email: tolga.goze@alcatel-lucent.com

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 that can result in approximately 1.5^n different piecewise linear models on the regressor space. For each region, 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. Our algorithm achieves this performance only with a computational complexity upper bounded by $O(n)$ in the worst-case and $O(\log(n))$ 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— Sequential, nonlinear, piecewise linear, regression, infinite depth context tree.

1. 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 paper, 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 $\{\boldsymbol{x}[t]\}_{t\geq 1}$. We refrain from any statistical assumptions on the unknown desired signal $\{d[t]\}_{t\geq 1}$ and the regressor vectors $\{\boldsymbol{x}[t]\}_{t\geq 1}$, where the desired sequence and the regressor vectors are real valued and bounded, i.e., $d[t] \in \mathbb{R}$, $\boldsymbol{x}[t] \triangleq [x_1[t], \ldots, x_p[t]]^T \in \mathbb{R}^p$ for an arbitrary integer p and $|d[t]|, |x_i[t]| < A < \infty$ for all t and $i = 1, \ldots, p$. 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 $d[1], \ldots, d[t-1]$ and the observed regressor vectors, $\boldsymbol{x}[1], \ldots, \boldsymbol{x}[t]$. That is, say we have[1] a sequential regressor $\hat{d}[t] = f(\boldsymbol{x}[t])$, then the regressor function $f(\cdot)$ can be constructed using only d_1^{t-1} and $\boldsymbol{x}_1^t$.

A simple and widely used regressor function is the linear regressor $f(\boldsymbol{x}[t]) = \boldsymbol{w}_t^T \boldsymbol{x}_t$, where the weighting parameter $\boldsymbol{w}_t$ 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 nonlinearity 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, is linear in the data length n, i.e., $O(n)$.

Hence, the main contributions of this paper 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

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

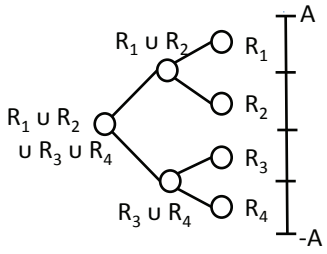

Fig. 1. The partitioning of a one dimensional regressor space, i.e., $[-A, A]$, using a depth-2 full context tree.

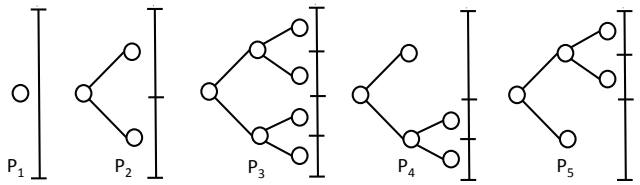

Fig. 2. All different piecewise linear models that can be obtained using a depth-2 full context tree, where the regressor space is one dimensional. These models are based on the partitioning shown in Fig. 1.

performance converges to 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.

2. 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 $\mathcal{R}_1, \ldots, \mathcal{R}_K$ 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 linear[2] regressor predicts $d[t]$ as $\hat{d}_m[t] = \boldsymbol{v}_{m,k}^T[t]\boldsymbol{x}[t]$ when $\boldsymbol{x}[t] \in \mathcal{R}_k$, where $\boldsymbol{v}_{m,k}[t] \in \mathbb{R}^p$ for all $k = 1, \ldots, K$.

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., $\boldsymbol{v}_{m,k}[t]$, 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].

As an example, in Fig. 1, we partition the one dimensional regressor space using a depth-2 tree, where the regions $\mathcal{R}_1, \ldots, \mathcal{R}_4$ 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

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

for any n, where $\mathcal{M}$ denotes the set of all different piecewise linear models defined on the infinite depth context tree, $\boldsymbol{v}_{m,k}$ is the regression parameter of the kth partition of the mth piecewise linear model such that $\hat{d}_b[t] = \boldsymbol{v}_{m,k}^T \boldsymbol{x}[t]$ is the prediction of a batch regressor (when $x[t] \in \mathcal{R}_k$), 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.

3. 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 $\{\boldsymbol{x}[t]\}_{t\geq 1}$ be arbitrary, bounded, and real-valued sequences of data and regressor vectors, respectively. Then the algorithm $\hat{d}[t]$ given in Section 3.1 when applied to*

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

these data sequences yields

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

for any n, with a computational complexity upper bounded by $O(n)$, where $\mathcal{M}'\triangleq\{m\in\mathcal{M}:K_m\leq O(\log(n))\}$, $\boldsymbol{v}_m\triangleq[\boldsymbol{v}_{m,1};\ldots;\boldsymbol{v}_{m,K_m}]$, and K_m represents the number of disjoint regions in model m.

This theorem implies that our algorithm given in Section 3.1, 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.

3.1. Outline of the 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 mth model, having K_m disjoint regions $\mathcal{R}_1,\ldots,\mathcal{R}_{K_m}$ 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 $\hat{d}_m[t]=\boldsymbol{v}_{m,k}^T[t]\,\boldsymbol{x}[t]$, when $\boldsymbol{x}[t]\in\mathcal{R}_k$, with the corresponding regression parameters [13]. The upper bound on the performance of this regressor can be calculated following similar lines to [13] and it is obtained as follows

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

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 $\kappa=\nu_1\ldots\nu_l$ is a binary string of length l formed from letters $\nu_1,\ldots,\nu_l$, we label its upper and lower children as $\kappa1$ and $\kappa0$, 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 $\kappa'=\nu'_1\ldots\nu'_{l'}$ is a prefix to string $\kappa=\nu_1\ldots\nu_l$ if $l'\leq l$ and $\nu'_i=\nu_i$ for all $i=1,\ldots,l'$, and the empty string λ is a prefix to all strings. Finally, we let $\mathcal{P}(\kappa)$ represent all prefixes to the string κ, i.e., $\mathcal{P}(\kappa)\triangleq\{\kappa_0,\ldots,\kappa_l\}$, 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 $i=0,\ldots,l$.

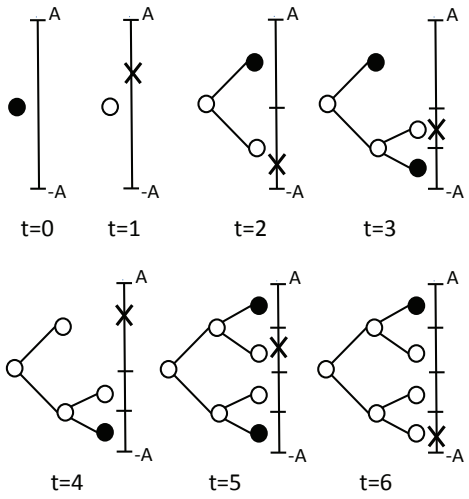

Fig. 3. A sample evolution of the infinite depth context tree, where the regressor space is one dimensional. The "$\times$" marks on the regressor space represents the value of the regressor vector at that specific time instant. Light nodes are the ones having an index of 1, whereas the index of the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

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 $\boldsymbol{x}_{\kappa,n}\triangleq\{\boldsymbol{x}[t],\,\forall t\in\{1,n\}:\boldsymbol{x}[t]\in\mathcal{R}_\kappa\}$.

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 $\boldsymbol{x}[t]\in\mathcal{R}_\kappa$. For this node if we have

- $\alpha_\kappa=1$, then we generate two children nodes $\kappa0,\kappa1$ for this node by dividing the region $\mathcal{R}_\kappa$ into two disjoint regions $\mathcal{R}_{\kappa0},\mathcal{R}_{\kappa1}$ 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 ith dimension. Then, we divide the information stored in $\boldsymbol{x}_{\kappa,n}$ into $\boldsymbol{x}_{\kappa0,n},\boldsymbol{x}_{\kappa1,n}$ and assign these sets to the nodes $\kappa0,\kappa1$, respectively. Using this information, we calculate $\boldsymbol{v}_{m,\kappa0}[t],\boldsymbol{v}_{m,\kappa1}[t]$ and finally set $\alpha_{\kappa\nu}=1$ for the node $\kappa\nu$, where $\nu\in\{0,1\}$, such that $\boldsymbol{x}[t]\in\mathcal{R}_{\kappa\nu}$, 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 $\times$'s 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 $\boldsymbol{x}[3] \in \mathcal{R}_0$ and divide this region into two disjoint regions using $x_1 = -A/2$ line. We then find that in fact $\boldsymbol{x}[3] \in \mathcal{R}_{01}$, hence set $\alpha_{01} = 1$, whereas $\alpha_{00} = 0$. 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

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

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

$$P_\kappa(n) \triangleq \frac{1}{2}P_{\kappa 0}(n)P_{\kappa 1}(n) + \frac{1}{2}\exp\left\{-\frac{1}{2a}\sum_{t\leq n\,:\,\boldsymbol{x}[t]\in\mathcal{R}_\kappa}\left(d[t]-\hat{d}_{m,k}[t]\right)^2\right\}.$$

After some algebra [10, 11], it can be shown that

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

where the first term follows due to the mixture-of-experts approach and the second term follows due to the adaptive construction of the infinite depth context tree. Using these node weights, we can construct a sequential algorithm [6], hence this concludes the proof of the theorem. □

Remark 1: By limiting the maximum depth of the tree by $O(\log(t))$ at each time t, we can 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. 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 $o(n/\log(n))$.

4. SIMULATIONS

In this section, we illustrate the performance of the introduced algorithm for the chaotic signal generated from the Duffing map. 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. We denote the infinite depth context tree algorithm of Theorem 1 by "IDT", the context tree weighting algorithm of [6] by "CTW", the linear regressor by "LR", the Volterra series regressor by "VSR" [14], and the sliding window Multivariate Adaptive Regression Splines of [15, 16] by "MARS".The combination weights of the LR and VSR are updated using the recursive least squares (RLS) algorithm [9]. The CTW algorithm has depth 2, the VSR and MARS algorithms are second order, and the MARS algorithm uses 21 knots with a window length of 500 that shifts in every 200 samples.

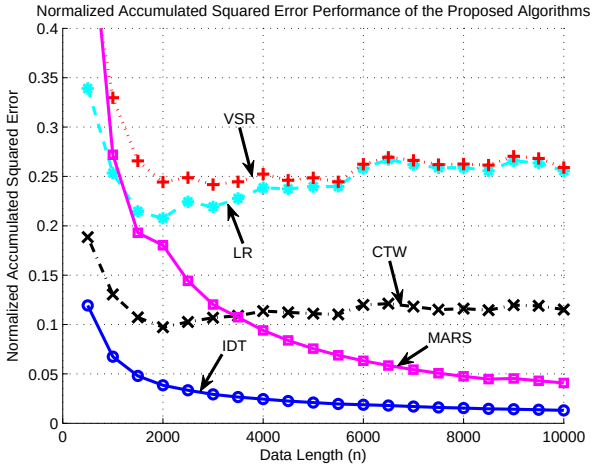

Fig. 4. Normalized cumulative squared error performances for the chaotic data generated by the Duffing map.

Fig. 4 shows the normalized cumulative squared error performances of the proposed algorithms. Since the conventional nonlinear and piecewise linear regression algorithms commit to a priori partitioning and/or basis functions, their performances are limited by the performances of the optimal batch regressors using these prior partitioning and/or basis functions as can be observed in Fig. 4. Hence, such prior selections result in fundamental performance limitations for these algorithms. For example, in the CTW algorithm, the partitioning of the regressor space is set before the processing starts. If this partitioning does not match with the underlying partitioning of the regressor space, then the performance of the CTW algorithm becomes highly unsatisfactory as seen in Fig. 4. Unlike such nonlinear models, the introduced algorithm does not commit to any prior structure and basis functions, instead it increments the number of disjoint regions to increase its nonlinear modeling power as the observed data length increases.

5. 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 best piecewise linear model defined on the infinite depth context tree. Furthermore, this performance is achieved only with a computational complexity $O(\log(n))$ under certain regularity conditions. We demonstrate performance gains of the introduced algorithm over a prediction scenario of a chaotic signal.

6. 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, Mar 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, Mar 1998.

[3] I. Ali and Y.-T. Chen, "Design quality and robustness with neural networks," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1518–1527, Nov 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, May 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, April 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, Mar 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] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. NJ: John Wiley & Sons, 1980.

[15] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67, 1991.

[16] ——, "Fast MARS," *Stanford University Technical Report*, 1993. [Online]. Available: http://www.milbo.users.sonic.net/earth/Friedman-FastMars.pdf