On the Statistical Efficiency of Compositional Nonparametric Prediction

Yixi Xu
Department of Statistics
Purdue University
West Lafayette, IN 47907, USA

Jean Honorio
Department of Computer Science
Purdue University
West Lafayette, IN 47907, USA

Xiao Wang
Department of Statistics
Purdue University
West Lafayette, IN 47907, USA

Abstract

In this paper, we propose a compositional nonparametric method in which a model is expressed as a labeled binary tree of $2k+1$ nodes, where each node is either a summation, a multiplication, or the application of one of the q basis functions to one of the p covariates. We show that in order to recover a labeled binary tree from a given dataset, the sufficient number of samples is $O(k\log(pq)+\log(k!))$, and the necessary number of samples is $\Omega(k\log(pq)-\log(k!))$. We further propose a greedy algorithm for regression in order to validate our theoretical findings through synthetic experiments.

1 Introduction

Nonparametric methods, such as spline-based methods and kernel-based methods, have been widely used in the past 20 years. Most existing methods make assumptions regarding the structure of the model in terms of interactions. For instance, the work of [12] assumes an additive structure of the predictor function, while in [4] the kernel family is defined as polynomial combinations of base kernels of a fixed degree. On the one hand, there is usually insufficient evidence from the data to support the assumption of a specific structure. On the other hand, inclusion of all interactions especially of high order terms would be burdensome for computing especially when the data is high dimensional. A commonly used strategy is to only include low order interactions into the model [4]. However, this would still be a restrictive assumption.

Our goal is to discover the complex structure of the *predictor* function in a concise manner. In contrast, existing methods focus on the discovery of the structure of *kernels* [4,6]. As an illustrative example for predictor functions, consider the work of Schmidt et al. [14], which discovered physical laws from experimental data, and provided concise analytical expressions that are amenable to human interpretation.

We build our model by compositionally adding or multiplying basis functions applied to specific dimensions of the covariate. This model is structurally equivalent to a labeled binary tree. The sum-product structure has demonstrated its versatility for several problems. Examples include sum-product networks for computation of partition functions and marginals of high-dimensional distributions [10] and structure discovery in nonparametric regression for automatic selection of the kernel family [6].

Our model is a generalization of several popular methods. For illustration, consider the following examples:

- Tensor product spline surfaces [3]: Assume there are two covariates $\boldsymbol{x}=(x_1,x_2)$, and define $g(\boldsymbol{x})=\sum_{i=1}^{q}\sum_{j=1}^{q}\beta_{ij}\phi_i(x_1)\phi_j(x_2)$, given the basis functions $\phi_1,\ldots,\phi_q:\mathbb{R}\to\mathbb{R}$. For simplicity, assume $q=2$, then Figure 1(a) is one visualization of g, where $\beta_{11}=w_1w_3,\beta_{12}=w_1w_4,\beta_{21}=w_2w_3,\beta_{22}=w_2w_4$.
- Sparse additive models [12]: Assume that $g(\boldsymbol{x})$ has an additive decomposition, where $\boldsymbol{x}=(x_1,\ldots,x_p)$. Define $g(\boldsymbol{x})=\sum_{j=1}^{p}\phi_{a_j}(x_j)$, where $a_1,\ldots,a_p\in\{1,\ldots,q\}$ and such that $\sum_{j=1}^{p}\mathbb{I}(\phi_{a_j}\neq 0)\leq s$ for some integer $s\ll p$.
- Tensor decomposition: Given a set of q functions $\phi_1,\ldots,\phi_q$ and a tensor y_{ijk} for $i,j,k=1,\ldots,p$. The problem is to find the indices $a_r,b_r,c_r\in$

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84. Copyright 2018 by the author(s).

$\{1, \ldots, q\}$ for $r = 1, \ldots, R$, that minimize: $\sum_{i=1}^{p} \sum_{j=1}^{p} \sum_{k=1}^{p} \left(\sum_{r=1}^{R} w_r \phi_{a_r}(i) \phi_{b_r}(j) \phi_{c_r}(k) - y_{ijk} \right)^2$. Note that $\sum_{r=1}^{R} w_r \phi_{a_r}(i) \phi_{b_r}(j) \phi_{c_r}(k)$ can be written as a fixed weighted labeled binary tree. Figure 1(b) illustrates the case when $R = 2$.

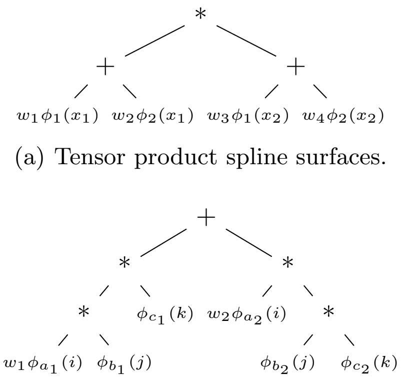
Our contribution is as follows. First, we propose a general compositional sum-product nonparametric method, in which a model is expressed as a weighted labeled binary tree. Second, we provide a generalization bound that holds for any data distribution and any weighted labeled binary tree. We show that $O(k \log(pq) + \log k!)$ samples are sufficient, by using Rademacher-complexity arguments. Third, we further show that $\Omega(k \log(pq) - \log k!)$ samples are necessary, by using information-theoretic arguments. Thus, our sample complexity bounds are tight. Furthermore, since the sample complexity is *logarithmic* in p and q, our method is statistically suitable for high dimensions and a large number of basis functions. Finally, we propose a well-motivated greedy algorithm for regression in order to validate our theoretical findings.

For comparison with results on sparse additive models, the work of [12] presents an L_1-regularization approach. Additionally, a sample complexity of $O(q \log((p - s)q))$ was shown to be sufficient for the correct identification of the basis functions in the sparse additive model. Note that in our work, we are interested in generalization bounds for the prediction error. The necessary number of samples for sparse additive models was analyzed in [11], where a sample complexity of $\Omega(s \log p)$ was found for the recovery of a function that is close to the true function in L_2-norm. Our sample complexity guarantee of $O(k \log p)$ matches this bound.

The paper is structured as follows. In Section 2, we provide a generalization bound. Section 3 discusses the necessary number of samples. In Section 4, we propose a greedy search algorithm for regression. In Section 5, we validate our theoretical results through synthetic experiments.

2 Compositional Nonparametric Trees for the General Prediction Problem

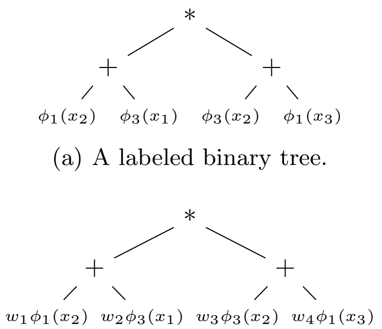
In this section, we define the general prediction problem, and then propose a solution via a compositional nonparametric method, in which a model is defined as a weighted labeled binary tree. In this tree, each node represents a multiplication, an addition, or the application of a basis function to a particular covariate.



(a) Tensor product spline surfaces.

(b) Tensor decomposition.

Figure 1: Examples of tensor product spline surfaces and tensor decomposition.



(a) A labeled binary tree.

(b) A weighted labeled binary tree.

Figure 2: Two tree examples.

The General Prediction Problem. Assume that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are n independent random variables on $\mathcal{X} = \mathbb{R}^p$, $y_1, \ldots, y_n$ are on $\mathcal{Y} \subseteq \mathbb{R}$. The general prediction problem is defined as

$$y_i = t(g(\boldsymbol{x}_i) + \epsilon_i), \tag{1}$$

where $t : \mathbb{R} \to \mathcal{Y}$ is a fixed function related to the prediction problem, $g : \mathbb{R}^p \to \mathbb{R}$ is an unknown function, and ϵ_i is an independent noise. We provide two examples in order to illustrate how to adopt equation (1) to different settings. For regression, we define $t(z) = z$, while for classification, we define $t(z) = sign(z)$.

The Labeled Binary Tree. We define a functional structure built compositionally by adding and multiplying a small number of basis functions. A straightforward visualization of this structure is a labeled binary tree. Given an infinite set of basis functions $\Phi = \{\phi_l, l = 1, 2, \cdots, \infty\}$ on $\mathbb{R} \to [-1, 1]$ and a truncation parameter q, $\mathcal{F}_{2k+1}$ is a set of binary trees where:

1. there are no more than $2k + 1$ nodes,

2. the labels of non-leaf nodes can be either "+" or "*",

3. the label of a leaf node can only be a function in Φ on a specific dimension of the covariate $\boldsymbol{x} = (x_1, \ldots, x_p)$, that is $\phi_i(x_j)$ for any $i = 1, \ldots, q$ and $j = 1, \ldots, p$,

Figure 2(a) gives an example of a labeled binary tree with seven nodes. All the leaves are $\phi_i(x_j)$s, while all non-leaf nodes are operations. Note that if we switch the left sub-tree and the right sub-tree, we obtain an equivalent structure.

As pointed out later in Remark 1, in the nonparametric setting, both k and q are allowed to grow as a function of n.

The Weighted Labeled Binary Tree. It is easy to show that a labeled binary tree with $2k+1$ nodes has the following properties:

1. It includes k operations.

2. It has $k+1$ leaves.

An easy way to add weights is to directly add weights to each leaf node, as shown in Figure 2(b). So given a tree structure $f \in \mathcal{F}_{2k+1}$, we can define $\mathcal{W}(f)$ as the set of all weighted labeled binary trees given f, with constraint $\|\boldsymbol{w}\|_1 \leq 1$. Additionally, we define

$$\mathcal{W}_{2k+1} = \bigcup_{f \in \mathcal{F}_{2k+1}} \mathcal{W}(f). \tag{2}$$

For a fixed $f \in \mathcal{F}_{2k+1}$, any $h \in \mathcal{W}(f)$ can be rewritten as a summation of some basis functions and some productions of basis functions. For instance, given $\boldsymbol{w}$ and the labeled binary tree structure f_0 in Figure 2(a), Figure 2(b) represents a function $h(x; f_0, \boldsymbol{w}) = (w_1\phi_1(x_2) + w_2\phi_3(x_1)) * (w_3\phi_3(x_2) + w_4\phi_1(x_3))$, and it is the summation of 4 interactions $w_1w_3\phi_1(x_2)\phi_3(x_2)$, $w_1w_4\phi_1(x_2)\phi_1(x_3)$, $w_2w_3\phi_3(x_1))\phi_3(x_2)$, and $w_2w_4\phi_3(x_1)\phi_1(x_3)$. Equivalently, $h(x; f_0, \boldsymbol{w}) = \langle \boldsymbol{v}, \boldsymbol{u} \rangle$, where $\boldsymbol{v} = \psi^v_{f_0}(\boldsymbol{w}) = (w_1w_3, w_1w_4, w_2w_3, w_2w_4)$ and $\boldsymbol{u} = \psi^u_{f_0}(\boldsymbol{x}) = (\phi_1(x_2)\phi_3(x_2), \phi_1(x_2)\phi_1(x_3), \phi_3(x_1)\phi_3(x_2), \phi_3(x_1)\phi_1(x_3))$. Similarly, for any labeled binary tree f, we could write $h = h(x; f, \boldsymbol{w}) \in \mathcal{W}(f)$ as an inner product of two vectors $\boldsymbol{v}$ and $\boldsymbol{u}$:

$$h(x; f, \boldsymbol{w}) = \langle \boldsymbol{v}, \boldsymbol{u} \rangle, \quad \boldsymbol{v} = \psi^v_f(\boldsymbol{w}), \quad \boldsymbol{u} = \psi^u_f(\boldsymbol{x}), \tag{3}$$

where the transformation function ψ^v_f and ψ^u_f depend on f. Define the length of the vector $\boldsymbol{v}$ and $\boldsymbol{u}$ as M_f, and M_f also depends on f. Define

$$M_{2k+1} = \max_{f \in \mathcal{F}_{2k+1}} M_f. \tag{4}$$

Lemma 1. *If $\|\boldsymbol{w}\|_1 \leq 1$ and $\|\phi_i\|_\infty \leq 1\ \forall i$, regardless of f, we always have $\|\boldsymbol{v}\|_1 \leq 1$ and $\|\boldsymbol{u}\|_\infty \leq 1$.*

Proof sketch. By induction. □

(Detailed proofs can be found on Appendix A.)

3 Sufficient Number of Samples

In this section, we provide a generalization bound that holds for any data distribution and any labeled binary tree. This not only implies the sufficient number of samples to recover a labeled binary tree from a given dataset, but also guarantees that the empirical risk (i.e., the risk with respect to a training set) is a consistent estimator of the true risk (i.e., the risk with respect to the data distribution). We first bound the size of $\mathcal{F}_{2k+1}$, and then show a Rademacher-based uniform convergence guarantee.

Properties of the Labeled Binary Tree Set. Let $|\mathcal{F}_{2k+1}|$ denote the size of $\mathcal{F}_{2k+1}$: the labeled binary tree set with no more than $2k+1$ nodes. The lemma below gives the upper bound of the size of the functional space, which will be used later to show the uniform convergence.

Lemma 2. *For $k \geq 1$, we have $|\mathcal{F}_{2k+1}| \leq 4k(k)!(pq)^{k+1}$.*

Proof sketch. By induction. □

The lemma below gives the upper bound of M_{2k+1}, which is used to later to bound the Rademacher complexity. Remind that M_{2k+1} is defined in eq.(4).

Lemma 3. $M_{2k+1} < (1.45)^{k+1}$.

Proof sketch. By induction. □

Rademacher-based Uniform Convergence. Next, we present our first main theorem, which guarantees a uniform convergence of the empirical risk to the true risk, regardless of the tree structure and weights.

Assume that $d : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ is a 1-Lipschitz function related to the prediction problem. For regression, we assume $\mathcal{Y} = \mathbb{R}$, and $d(y, y') = \min(1, (y - y')^2/2)$, while for classification, we assume $\mathcal{Y} = \{-1, 1\}$, and $d(y, y^{'}) = \min(1, \max(0, 1 - yy^{'}))$. Let $z = (x, y) \in \mathcal{Z}$, where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Furthermore, let $\mathcal{H}(f) = \{h(z) = d(y, g(x)), g \in \mathcal{W}(f)\}$ for a fixed labeled binary tree f. Let $\mathcal{H}_{2k+1}$ be a hypothesis class satisfying

$$\mathcal{H}_{2k+1} = \bigcup_{f \in \mathcal{F}_{2k+1}} \mathcal{H}(f).$$

For every $h \in \mathcal{H}(f)$, we define the true and empirical risks as

$$\mathbb{E}_{\mathcal{D}}[h] = \mathbb{E}_{z \sim \mathcal{D}}[h(z)], \quad \widehat{\mathbb{E}}_S[h] = \frac{1}{n}\sum_{i=1}^{n} h(z_i). \tag{5}$$

Next, we state our generalization bound that shows that $O(k \log(pq) + \log k!)$ samples are sufficient for learning.

Theorem 1. *Let $z = (x, y)$ be a random variable of support $\mathcal{Z}$ and distribution $\mathcal{D}$. Let $S = \{z_1 \dots z_n\}$ be a dataset of n i.i.d. samples drawn from $\mathcal{D}$. Fix $\delta \in (0, 1)$. With probability at least $1 - \delta$ over the choice of S, we have:*

$$\begin{aligned}&(\forall f \in \mathcal{F}_{2k+1}, \forall h \in \mathcal{H}(f))\\ &\mathbb{E}_{\mathcal{D}}[h] \leq \widehat{\mathbb{E}}_S[h] + 2\sqrt{\frac{k+1}{n}} +\\ &\sqrt{\frac{(k+1)\log pq + \log 8k(k)! + \log(1/\delta)}{2n}}\end{aligned}$$

Proof. Given a function $h : \mathcal{Z}^n \to \mathbb{R}$, we define $\mathbb{E}_S[h(S)] = \mathbb{E}_{S \sim \mathcal{D}^n}[h(S)]$. The function $\varphi_f(S) = \sup_{h \in \mathcal{H}(f)} \left(\mathbb{E}_{\mathcal{D}}[h] - \widehat{\mathbb{E}}_S[h]\right)$ fulfills the condition in McDiarmid's inequality and $\mathcal{H}(f) \subseteq \{h | h : \mathcal{Z} \to [0, 1]\}$, by Lemma 4 (Please see Appendix B.), therefore $\mathbb{P}[\varphi_f(S) - \mathbb{E}_S[\varphi_f(S)] \geq \varepsilon] \leq e^{\frac{-2\varepsilon^2}{\sum_{i=1}^{n}(1/n)^2}} = e^{-2n\varepsilon^2}$. Furthermore, by applying the union bound for all $f \in \mathcal{F}_{2k+1}$, by Lemma 2, and by Hoeffding's inequality, we have:

$$\begin{aligned}&\mathbb{P}[(\exists f \in \mathcal{F}_{2k+1}), \varphi_f(S) - \mathbb{E}_S[\varphi_f(S)] \geq \varepsilon]] \leq\\ &\sum_{f \in \mathcal{F}_{2k+1}} \mathbb{P}[\varphi_f(S) - \mathbb{E}_S[\varphi_f(S)] \geq \varepsilon] \leq 2|\mathcal{F}_{2k+1}|e^{-2n\epsilon^2}\\ &\leq 8k(k)!(pq)^{k+1}e^{-2n\epsilon^2}\end{aligned}$$

Equivalently, $\mathbb{P}[(\forall f \in \mathcal{F}_{2k+1}), \varphi_f(S) - \mathbb{E}_S[\varphi_f(S)] \leq \varepsilon]] \geq 1 - 8k(k)!(pq)^{k+1}e^{-2n\epsilon^2}$.

Setting $8k(k)!(pq)^{k+1}e^{-2n\epsilon^2} = \delta$, we get $\varepsilon = \sqrt{\frac{(k+1)\log pq + \log 8k(k)! + \log(1/\delta)}{2n}}$. Thus:

$$\begin{aligned}&\mathbb{P}\,[(\forall f \in \mathcal{F}_{2k+1}), \varphi_f(S) < \mathbb{E}_S[\varphi_f(S)] +\\ &\sqrt{\frac{(k+1)\log pq + \log 8k(k)! + \log(1/\delta)}{2n}}]\\ &\geq 1 - \delta\end{aligned} \tag{6}$$

Note that by the definition of the supremum, by the definition of the function $\varphi_f : \mathcal{Z}^n \to \mathbb{R}$, and by eq.(6), with probability at least $1 - \delta$, simultaneously for all $f \in \mathcal{F}_{2k+1}$ and $h \in \mathcal{H}(f)$

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[h] - \widehat{\mathbb{E}}_S[h] &\leq \sup_{h \in \mathcal{H}(f)} \left(\mathbb{E}_{\mathcal{D}}[h] - \widehat{\mathbb{E}}_S[h]\right)\\ &= \varphi_f(S)\\ &< \sqrt{\frac{(k+1)\log pq + \log 8k(k)! + \log(1/\delta)}{2n}} +\\ &\mathbb{E}_S[\varphi_f(S)]\end{aligned} \tag{7}$$

The next step is to bound $\mathbb{E}_S[\varphi_f(S)]$ in eq.(7) in terms of the Rademacher complexity of $\mathcal{W}(f)$. By the definition of φ_f, by the ghost sample technique, the Ledoux-Talagrand Contraction Lemma, we can show that

$$\mathbb{E}_S[\varphi_f(S)] = 2\mathfrak{R}_n(\mathcal{H}(f)) \leq 2\mathfrak{R}_n(\mathcal{W}(f))$$

The final step is to bound $\mathfrak{R}_n(\mathcal{W}(f))$, and it is sufficient to bound $\hat{\mathfrak{R}}_S(\mathcal{W}(f))$ for any $f \in \mathcal{F}_{2k+1}$. Then for a fixed $f \in \mathcal{F}_{2k+1}$, any $g \in \mathcal{W}(f)$ can be rewritten as a summation of no more than $[(1.45)^{k+1}]$ productions of basis functions, where $[m]$ denotes that largest integer smaller than or equal to m according to Lemma 3. We could decompose $h = h(x; f, \boldsymbol{w})$ as in equation (3), thus $h = h(\boldsymbol{x}; f, \boldsymbol{w}) = \langle \boldsymbol{v}, \boldsymbol{u} \rangle$, where $||\boldsymbol{v}||_1 \leq 1$ and $||\boldsymbol{u}||_\infty \leq 1$ by Lemma 1. By using a technique similar to [9] for linear prediction, we have

$$\begin{aligned}\hat{\mathfrak{R}}_S(\mathcal{W}(f)) &= \mathbb{E}_\sigma\left[\sup_{g \in \mathcal{W}(f)}\left(\frac{1}{n}\sum_{i=1}^{n}\sigma_i g(\boldsymbol{x}^{(i)})\right)\right]\\ &= \mathbb{E}_\sigma\left[\sup_{\|\boldsymbol{w}\|_1 \leq 1}\left(\frac{1}{n}\sum_{i=1}^{n}\sigma_i g(\boldsymbol{x}^{(i)}; \boldsymbol{w}, f)\right)\right]\\ &\leq \frac{1}{n}\ \mathbb{E}_\sigma\left[\sup_{\|\boldsymbol{v}\|_1 \leq 1}\left(\sum_{i=1}^{n}\sigma_i\langle \boldsymbol{v}, \boldsymbol{u}^{(i)}\rangle)\right)\right]\\ &= \frac{1}{n}\ \mathbb{E}_\sigma\left[\sup_{\|\boldsymbol{v}\|_1 \leq 1}\langle \boldsymbol{v}, \textstyle\sum_{i=1}^{n}\sigma_i \boldsymbol{u}^{(i)}\rangle\right]\\ &= \frac{\|\boldsymbol{v}\|_1}{n}\ \mathbb{E}_\sigma\left[\|\textstyle\sum_{i=1}^{n}\sigma_i \boldsymbol{u}^{(i)}\|_\infty\right]\\ &= \frac{1}{n}\ \mathbb{E}_\sigma\left[\sup_j \textstyle\sum_{i=1}^{n}\sigma_i[\boldsymbol{u}^{(i)}]_j\right]\\ &= \frac{\sqrt{2\log M_{2k+1}}}{n}\ \sup_j\sqrt{\textstyle\sum_{i=1}^{n}[\boldsymbol{u}^{(i)}]_j^2}\\ &\leq \frac{\sqrt{2\log M_{2k+1}}}{n}\ \sqrt{n\|\boldsymbol{u}\|_\infty^2}\\ &\leq \sqrt{\frac{2\log M_{2k+1}}{n}}\\ &\leq \sqrt{\frac{2(k+1)\log 1.45}{n}}\\ &< \sqrt{\frac{k+1}{n}}\end{aligned}$$

Finally, we have $\mathfrak{R}_n(\mathcal{W}(f)) = \mathbb{E}_{S \sim \mathcal{D}^n}[\hat{\mathfrak{R}}_S(\mathcal{W}(f))] < \sqrt{\frac{k+1}{n}}$ □

Corollary 1. *Define* $\hat{h} = \underset{h\in\mathcal{H}_{2k+1}}{\arg\min}\,\widehat{\mathbb{E}}_S[h]$, *and* $\bar{h} = \underset{h\in\mathcal{H}_{2k+1}}{\arg\min}\,\mathbb{E}_{\mathcal{D}}[h]$. *Then under the same setting of Theorem 1, fix* $\delta \in (0,1)$. *With probability at least* $1-2\delta$ *over the choice of* S, *we have:*

$$\mathbb{E}_{\mathcal{D}}[\hat{h}] - \mathbb{E}_{\mathcal{D}}[\bar{h}] \leq 2\sqrt{\frac{k+1}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}} + \sqrt{\frac{(k+1)\log pq + \log 8k(k)! + \log(1/\delta)}{2n}}$$

Proof. By Theorem 1, with probability at least $1-\delta$ over the choice of S,

$$\mathbb{E}_{\mathcal{D}}[\hat{h}] \leq \widehat{\mathbb{E}}_S[\hat{h}] + 2\sqrt{\frac{k+1}{n}} + \sqrt{\frac{(k+1)\log pq + \log 8k(k)! + \log(1/\delta)}{2n}}$$

By Hoeffding's inequality, with probability at least $1-\delta$ over the choice of S,

$$\widehat{\mathbb{E}}_S[\bar{h}] - \mathbb{E}_{\mathcal{D}}[\bar{h}] \leq \sqrt{\frac{\log(1/\delta)}{2n}}$$

Since $\hat{h}$ minimizes $\widehat{\mathbb{E}}_S[h]$, $\widehat{\mathbb{E}}_S[\hat{h}] \leq \hat{\mathbb{E}}_S[\bar{h}]$. With probability at least $1-2\delta$ over the choice of S,

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[\hat{h}] - \mathbb{E}_{\mathcal{D}}[\bar{h}] &= \mathbb{E}_{\mathcal{D}}[\hat{h}] - \widehat{\mathbb{E}}_S[\bar{h}] + \widehat{\mathbb{E}}_S[\bar{h}] - \mathbb{E}_{\mathcal{D}}[\bar{h}] \\ &\leq \mathbb{E}_{\mathcal{D}}[\hat{h}] - \widehat{\mathbb{E}}_S[\hat{h}] + \widehat{\mathbb{E}}_S[\bar{h}] - \mathbb{E}_{\mathcal{D}}[\bar{h}] \\ &\leq 2\sqrt{\frac{k+1}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}} + \sqrt{\frac{(k+1)\log pq + \log 8k(k)! + \log(1/\delta)}{2n}}\end{aligned}$$

□

Next, we present a useful remark in the nonparametric setting, where both k and q are allowed to grow as a function of n.

Remark 1. *If* $k \in O(\min(n^{1/2-\epsilon}, \frac{n^{1-2\epsilon}}{\log p}))$, $q \in O(e^{n^{1/2-\epsilon}})$ *for any* $\epsilon \in (0,1/2)$, *then the generalization error in Theorem 1 could be uniformly bounded by* $O(n^{-\epsilon})$.

4 Necessary Number of Samples

In this section, we analyze the necessary number of samples to recover a labeled binary tree from a given dataset. To show the necessary number of samples, we restrict the operation to multiplications only, and consider unit weights. Note that the necessary number of samples in restricted ensembles yields a lower bound for the original problem. The use of restricted ensembles is customary for information-theoretic lower bounds [13, 15]. We utilize Fano's inequality as the main proof technique.

We construct a restricted ensemble as follows. Define a sequence of basis functions $\phi_i(z) = \sqrt{2}\cos(i\pi z)$, where $z \in [-1,1]$ for $i = 1,\dots,q$. Furthermore, let $\boldsymbol{x}_i \sim Unif[-1,1]^p$, $\epsilon_i \sim N(0,\sigma_\epsilon^2)$. Let $S = \{(\boldsymbol{x}_i, z_i) : z_i = g(\boldsymbol{x}_i) + \epsilon_i, i = 1,\dots,n\}$, and $S^{'} = \{(\boldsymbol{x}_i, y_i) : y_i = t(z_i), i = 1,\dots,n\}$, where $t : \mathbb{R} \to \mathcal{Y}$ is a fixed function related to the prediction problem, as introduced in Section 2. This defines a Markov chain $g \to S \to S' \to \hat{g}$. To apply Fano's inequality, we need to further bound the mutual information $\mathbb{I}(g, S^{'})$ by a sum of Kullback-Leibler (KL) divergences of the form $KL(P_{\boldsymbol{x},y|g_i}|P_{\boldsymbol{x},y|g_i^{'}})$ where g_i and g_i' are two different compositional trees. Consider a labeled binary tree subspace $\mathcal{G}_{2k+1}$ of $\mathcal{F}_{2k+1}$, where we only allow for multiplication nodes (i.e., additions are not allowed) and where each covariate x_j of the independent variable $\boldsymbol{x}$ is used only once. Furthermore, we consider a restricted ensemble with unit weights. Equivalently,

$$\begin{aligned}\mathcal{G}_{2k+1} = \{g_{\mathcal{A}}(\boldsymbol{x}) = \prod_{(i,j)\in\mathcal{A}} \phi_i(x_j) :& \\ \mathcal{A} \subseteq \{1,\dots,q\}\times\{1,\dots,p\},& \\ |\mathcal{A}| \leq k+1, \forall(i,j)\in\mathcal{A},\ l \neq i \Rightarrow (l,j) \notin \mathcal{A}\}.&\end{aligned}$$

Let $c = |\mathcal{G}_{2k+1}| = \sum_{i=1}^{k} q^{i+1}\binom{p}{i+1}$.

Next, we state our information-theoretic lower bound that shows that $\Omega(k\log(pq) - \log k!)$ samples are necessary for learning.

Theorem 2. *Assume nature uniformly picks a true hypothesis* $\bar{g}$ *from* $\mathcal{G}_{2k+1}$. *For any estimator* $\hat{g}$, *if* $n \leq (\log(q^{k+1}\binom{p}{k+1})) - 2\log 2)\sigma_\epsilon^2/2$, *then* $\mathbb{P}[\hat{g} \neq \bar{g}] \geq \frac{1}{2}$.

Proof. Any $g_{\mathcal{A}} \in \mathcal{G}_{2k+1}$ can be decomposed by the dimension of x:

$$g_{\mathcal{A}}(\boldsymbol{x}) = \prod_{j=1}^{p} g_j^{\mathcal{A}}(x_j),$$

where $g_j^{\mathcal{A}} = \phi_{i_j}$ if $\exists(i_j,j)\in\mathcal{A}$, and $g_j^{\mathcal{A}} \equiv 1$ if $(i,j)\notin\mathcal{A}$ for any i. In addition, $\int_{-1}^{1}\frac{1}{2}\phi_i(x)dx = 0$ and $\langle\phi_i,\phi_{i'}\rangle = \int_{-1}^{1}\frac{1}{2}\phi_i(x)\phi_{i'}(x)dx = I(i = i^{'})$. Thus,

$$\begin{aligned}\langle g_{\mathcal{A}}, g_{\mathcal{A}'}\rangle &= \int_{-1}^{1}\cdots\int_{-1}^{1}\frac{1}{2^p}g_j^{\mathcal{A}}(x_j)g_j^{\mathcal{A}^{'}}(x_j)dx_1\cdots dx_p \\ &= \prod_{j=1}^{p}\int_{-1}^{1}\frac{1}{2}g_j^{\mathcal{A}}(x_j)g_j^{\mathcal{A}^{'}}(x_j)dx_j \\ &= \prod_{j=1}^{p} I(g_j^{\mathcal{A}} = g_j^{\mathcal{A}^{'}}) \\ &= I(g_{\mathcal{A}} = g_{\mathcal{A}'})\end{aligned}$$

Furthermore,

$$\begin{aligned} ||g_{\mathcal{A}} - g_{\mathcal{A}'}||^2 &= \langle g_{\mathcal{A}}, g_{\mathcal{A}}\rangle + \langle g_{\mathcal{A}'}, g_{\mathcal{A}'}\rangle - 2\langle g_{\mathcal{A}}, g_{\mathcal{A}'}\rangle \\ &= 2I(g_{\mathcal{A}} = g_{\mathcal{A}'}) \end{aligned} \tag{10}$$

By the data processing inequality [5] in the Markov chain $g \to S \to S' \to g$, and since the mutual information can be bounded by a pairwise KL bound [16], we have

$$\begin{aligned} \mathbb{I}(\bar{g}, S^{'}) &\leq \mathbb{I}(\bar{g}, S) \\ &\leq \frac{1}{c^2} \sum_{\mathcal{A}} \sum_{\mathcal{A}'} KL(P_{S|g_{\mathcal{A}}} | P_{S|g_{\mathcal{A}'}}) \\ &= \frac{n}{c^2} \sum_{\mathcal{A}} \sum_{\mathcal{A}'} KL(P_{\boldsymbol{x},y|g_{\mathcal{A}}} | P_{\boldsymbol{x},y|g_{\mathcal{A}'}}) \\ &= \frac{n}{c^2} \sum_{\mathcal{A}} \sum_{\mathcal{A}'} KL(\mathcal{N}(g_{\mathcal{A}}, \sigma_\epsilon^2) | \mathcal{N}(g_{\mathcal{A}'}, \sigma_\epsilon^2)) \\ &= \frac{n}{c^2} \sum_{\mathcal{A}} \sum_{\mathcal{A}'} \frac{||g_{\mathcal{A}} - g_{\mathcal{A}'}||^2}{2\sigma_\epsilon^2} \\ &\leq \frac{n}{c^2} * c^2 * \frac{2}{2\sigma_\epsilon^2} \\ &= \frac{n}{\sigma_\epsilon^2} \end{aligned}$$

By the Fano's inequality [5] on the Markov chain $g \to S \to S^{'} \to \hat{g}$, we have

$$\mathbb{P}[\hat{g} \neq \bar{g}] \geq 1 - \frac{\mathbb{I}(\bar{g}, S^{'}) + \log 2}{\log c} \geq 1 - \frac{n/\sigma_\epsilon^2 + \log 2}{\log c}$$

By making

$$\frac{1}{2} = \mathbb{P}[\hat{g} \neq \bar{g}] \geq 1 - \frac{n/\sigma_\epsilon^2 + \log 2}{\log c},$$

we have

$$n \leq (\log c - 2\log 2)\sigma_\epsilon^2/2$$

Since $c \geq q^{k+1}\binom{p}{k+1}$, $n \leq (\log(q^{k+1}\binom{p}{k+1})) - 2\log 2)\sigma_\epsilon^2/2$ implies $\mathbb{P}[\hat{g} \neq \bar{g}] \geq \frac{1}{2}$. If $p \gg k$, the above is equivalent to

$$\begin{aligned} n &= \Omega\left(\frac{\sigma_\epsilon^2}{2}(\log[q^{k+1}p^{k+1}/(k+1)!] - 2\log 2)\right) \\ &\in \Omega\left((k+1)\log(pq) - \log(k+1)!\right) \end{aligned}$$

□

Corollary 2. *Assume nature uniformly picks a true function $\bar{g}$ from $\mathcal{G}_{2k+1}$. For each $g \in \mathcal{G}_{2k+1}$, define a corresponding $h(\boldsymbol{x}, y) = \frac{1}{2}(y - g(\boldsymbol{x}))^2$. The corresponding true hypothesis is $\bar{h} = \bar{h}(\boldsymbol{x}, y) = \frac{1}{2}(y - \bar{g}(\boldsymbol{x}))^2$. Let $\mathcal{H}_{2k+1} = \{h(\boldsymbol{x}, y) = \frac{1}{2}(y - g(\boldsymbol{x}))^2, g \in \mathcal{G}_{2k+1}\}$. For any estimator $\hat{h} = \hat{h}(\boldsymbol{x}, y) = \frac{1}{2}(y - \hat{g}(\boldsymbol{x}))^2$, if $n \leq (\log(q^{k+1}\binom{p}{k+1})) - 2\log 2)\sigma_\epsilon^2/2$, then $\mathbb{E}_{\mathcal{D}}[\hat{h}] - E_{\mathcal{D}}[\bar{h}] \geq 1$ with probability at least $\frac{1}{2}$.*

Proof. $\bar{g}$ is the true function, so $y = \bar{g}(\boldsymbol{x}) + \epsilon$, where $\epsilon \sim N(0, \sigma_\epsilon^2)$. Recall that by Theorem 2, if $n \leq (\log(q^{k+1}\binom{p}{k+1})) - 2\log 2)\sigma_\epsilon^2/2$ then $P[\bar{g} \neq \hat{g}] \geq 1/2$. Thus, assuming that $\bar{g} \neq \hat{g}$, we have

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[\hat{h}] - E_{\mathcal{D}}[\bar{h}] &= \frac{1}{2}\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[(y - \hat{g}(\boldsymbol{x}))^2 - (y - \bar{g}(\boldsymbol{x}))^2] \\ &= \frac{1}{2}\mathbb{E}_{\substack{\boldsymbol{x}\sim Unif[-1,1]^p \\ \epsilon\sim N(0,\sigma_\epsilon^2)}}[(\bar{g}(\boldsymbol{x}) + \epsilon - \hat{g}(\boldsymbol{x}))^2 - \epsilon^2] \\ &= \frac{1}{2}\mathbb{E}_{\boldsymbol{x},\epsilon}[(\bar{g}(\boldsymbol{x}) - \hat{g}(\boldsymbol{x}))^2 + 2\epsilon(\bar{g}(\boldsymbol{x}) - \hat{g}(\boldsymbol{x}))] \\ &= \frac{1}{2}\mathbb{E}_{\boldsymbol{x}}[(\bar{g}(\boldsymbol{x}) - \hat{g}(\boldsymbol{x}))^2] + \\ &\quad \mathbb{E}_\epsilon[\epsilon] * \mathbb{E}_{\boldsymbol{x}}[(\bar{g}(\boldsymbol{x}) - \hat{g}(\boldsymbol{x}))] \\ &= \frac{1}{2}||\bar{g} - \hat{g}||^2 \\ &= \frac{1}{2} * 2I(\bar{g} \neq \hat{g}) \\ &= 1 \end{aligned}$$

□

Remark 2. *Excess risk measures how well the empirical risk minimizer performs when compared to the best candidate in the hypothesis class. On the one hand, Corollary 1 discusses the upper bound of the excess risk, and indicates that the sufficient sample complexity is $O(k\log(pq) + \log k!)$. On the other hand, Corollary 2 discusses the lower bound of the excess risk, and shows that the necessary sample complexity is $\Omega(k\log(pq) - \log k!)$. Especially when $k \ll pq$, both the sufficient sample complexity and necessary sample complexity are $\Theta(k\log(pq))$.*

5 Greedy Search Algorithm for Regression

In this section, we propose a greedy search algorithm to recover a weighted labeled binary tree for regression. As mentioned in Section 3.2, for regression, we define $d(y, y') = \min(1, (y - y')^2/2)$. For simplicity, we assume $\mathcal{Y} = [-1, 1]$, thus $d(y, y') = (y - y')^2/2$. Consequently, we have $\mathcal{H}(f) = \{h(z) = h(x, y) = (y - g(x))^2/2, g \in \mathcal{W}(f)\}$ for a fixed labeled binary tree f. The true risk and the empirical risk are defined as $\mathbb{E}_{\mathcal{D}}[h] = \mathbb{E}_{(x,y)\sim\mathcal{D}}[(y - g(x))^2/2]$, and $\hat{\mathbb{E}}_S[h] = \sum_{i=1}^{n}(y_i - g(\boldsymbol{x}_i))^2/2$.

Based on Theorem 1 in Section 3.2, it is straightforward to have a brute-force algorithm to traverse all possible trees in $\mathcal{F}_{2k+1}$, and to compute the best weights for each tree. Theorem 1 could guarantee that the risk at the empirical risk minimizer is close to the minimum possible risk over all functions in $\mathcal{W}_{2k+1}$, given enough training samples. However the space of trees grows exponentially with the number of nodes, as shown in

Lemma 2, and therefore the brute-force algorithm is exponential-time.

After decades of work, the literature in tensor decomposition has still failed to provide polynomial-time algorithms with guarantees, for a general nonsymmetric tensor decomposition problem. In general, it has been shown that most tensor problems are NP-hard [8]. Therefore most existing literature considers a specific tensor structure like the symmetric orthogonal decomposition [1]. As shown in Figure 1(b), we can model the tensor decomposition problem in our framework, for a fixed tree. However in our problem, we learn the tree structure. Thus, our problem is harder than tensor decomposition.

Given the above, we propose a greedy search algorithm for learning the structure of *predictor functions*. A greedy approach was also taken in [6] for learning the structure of *kernels*. Before we proceed, note that the uniform convergence of the empirical risk to the true risk holds for any $h \in \mathcal{H}_{2k+1}$ and therefore, it applies to the greedy algorithm output, which is an element of $\mathcal{H}_{2k+1}$.

Our algorithm begins by applying all basis functions to all input dimensions, and picking the one that minimizes $\sum_{m=1}^{n}(y_m - w'\phi_{i'}(x_{j'}))^2/2$ among all function indices $i' \in \{1, \dots, q\}$ and coordinates $j' \in \{1, \dots, p\}$, where w' is estimated separately for each candidate option (i', j'). This produces a tree with a single node. After this, we repeat the following search operators over the leaves of the current tree: Any leaf $\mathcal{V}$ can be replaced with $\mathcal{V} + \mathcal{V}^{'}$, or $\mathcal{V} * \mathcal{V}^{'}$, where $\mathcal{V}^{'} = w'\phi_{i'}(x_{j'})$.

Our algorithm searches over the space of trees using a greedy search approach. At each stage, we evaluate the replacement of every leaf by either a summation or multiplication, and compute the weight for the new candidate leaf while fixing all the other weights. Then we take the search operation with the lowest score among all leaves, and adjust all weights by coordinate descent at each iteration. (For completeness, we include our main algorithm in Appendix C.)

Computing the Weight. A main step in our main algorithm is the computation of the weight of a new candidate leaf, while fixing all the other weights. Fortunately, computing the new weight turns out to be a simple least square problem, but involves traversing the tree from the root to the candidate node being evaluated. (The corresponding algorithm can be found on Appendix C, with a concrete example to illustrate our algorithm.)

Computational Complexity. Next, we analyze the time complexity of our method. In iteration D, we solve $O(pqD)$ single-dimensional closed-form optimization problems: for all the D tree leaves, our algorithm tries to insert a new node with either ”+” or ”*”, all q basis functions, and all p dimensions of $\boldsymbol{x}$. In addition, it takes $O(nD)$ time to compute the optimal weight (in closed-form) for a specific basis function of a specific dimension of $\boldsymbol{x}$ at a specific insert position on a dataset of size n. Finally, it takes $O(nD)$ to adaptively update all weights at each step by coordinate descent. The computational complexity of our algorithm for k iterations is thus $O(pqn(1^2 + 2^2 + \cdots + k^2)) \in O(pqnk^3)$. This can be reduced by processing the tree leaves (or alternatively, batches of data samples) in parallel.

6 Experiments

In this section, we demonstrate our theorem in four simulation experiments. We use a function $g(\boldsymbol{x}) = 0.3sin(3\pi x_1)cos(2\pi x_2) + 0.4x_3^2 - 0.3x_4$, and noise standard deviation $\sigma = 0.05$. Our choice of the set of basis functions Φ include B-spline of degree 1, Fourier basis functions: $\{\sin(i\pi x), \cos(i\pi x)\}_{i=1,\dots,\infty}$ and truncated polynomials: $\{x, x^2, x^3, (x-t)_+^3, t \in \mathbb{R}\}$, where $(x)_+ = max(x, 0)$. We designed four different experiments to demonstrate our theoretical contributions. For each setting, the generalization error is estimated by the mean of 20 repeated trials in order to show error bars at 95% confidence level.

Experiment 1 We set the dimension of the explanatory variables $p = 100$, the number of basis functions: $q = 40$, and the number of iterations $k = 10$. For each value of $n \in \{50, 100, 150, 200, 250\}$, we sampled n random samples $\boldsymbol{x}_i$, $y_i = g(\boldsymbol{x}_i) + \epsilon_i$, $i = 1, \cdots, n$ for training, and $n/3$ samples for testing. In Figure 3, we observe that the generalization error has a sharp decline when n increases from 50 to 100, and a slower decline for higher values of n. This demonstrates that the generalization error $\propto \sqrt{\frac{1}{n}}$ as prescribed by Theorem 1.

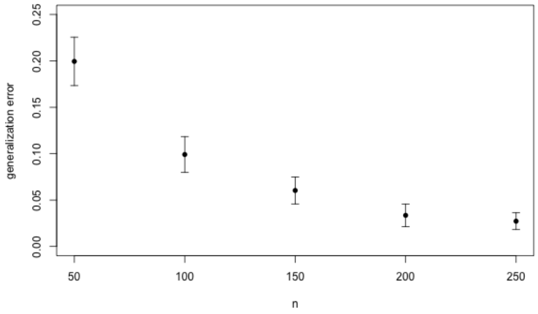


Figure 3: Generalization error vs. sample size n.

Experiment 2 We set the sample size $n = 250$, the number of basis functions: $q = 40$, and the number of iterations $k = 10$. For each value of $p \in \{10, 20, 50, 100, 200\}$, we sampled 250 $p-dimensional$ random samples $\boldsymbol{x}_i$, $y_i = g(\boldsymbol{x}_i) + \epsilon_i$, $i = 1, \cdots, n$ for training, and 83 samples for testing. Figure 4 shows that the generalization error grows rapidly when $p \in (0, 50)$, and the growth slows down as p increases. This finding matches the conclusion of Theorem 1 that the generalization error $\propto \sqrt{\log p}$.

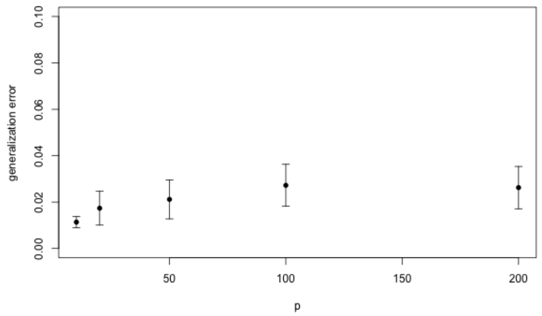


Figure 4: Generalization error vs. dimension of the explanatory variable p.

Experiment 3 We set the dimension of the explanatory variables $p = 100$, the number of basis functions: $q = 40$, and the sample size $n = 250$. For each value of the number of iterations $k \in \{1, 5, 10, 20\}$, we sampled 250 random samples $\boldsymbol{x}_i$, $y_i = g(\boldsymbol{x}_i) + \epsilon_i$, $i = 1, \cdots, n$ for training, and 83 samples for testing. As shown in Figure 5, the generalization error grows almost linearly as k increases when k is small, but the growth rate decreases apparently when $k > 15$. This is consistent with the theoretical result that the generalization error $\propto \sqrt{k}$.

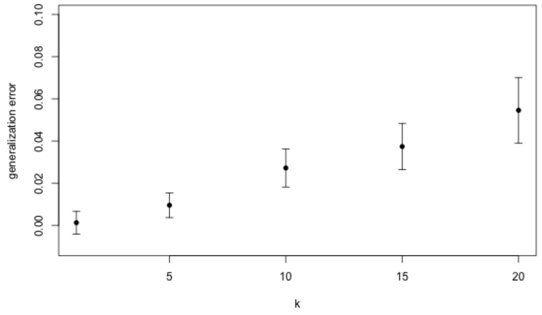


Figure 5: Generalization error vs. number of iterations k.

Experiment 4 We set the dimension of the explanatory variables $p = 20$, the sample size $n = 250$, and the number of iterations $k = 10$. For each value of $q \in \{10, 20, 50, 100\}$, we sampled 250 random samples $\boldsymbol{x}_i$, $y_i = g(\boldsymbol{x}_i) + \epsilon_i$, $i = 1, \cdots, n$ for training, and 83 samples for testing. Figure 6 indicates that the generalization error grows rapidly when q is small, and the growth slows down as q continue to increase. This matches the conclusion of Theorem 1 that the generalization error $\propto \sqrt{\log q}$.

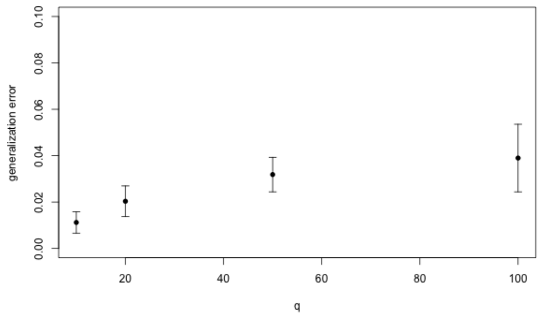


Figure 6: Generalization error vs. number of basis functions q.

Our methods are comparative to methods like *Gaussian processes* for two real-world data sets, although our model sizes are much smaller. (Please see Appendix D.)

7 Concluding Remarks

There are several ways of extending this research. While we focused on the sample complexity for trees of predictor functions, it would be interesting to analyze trees of kernels as well, as many popular kernel structures [6] are equivalent to a labeled binary tree. Additionally, while we focused on learning trees, it would be interesting to propose methods for learning general directed acyclic graphs.

References

[1] Animashree Anandkumar, Rong Ge, Daniel J Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[2] P. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[3] Carl De Boor. *A practical guide to splines*, volume 27. Springer-Verlag New York, 1978.

[4] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Learning non-linear combinations of kernels. In *Advances in Neural Information Processing Systems*, pages 396–404, 2009.

[5] T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley & Sons, 2nd edition, 2006.

[6] David K Duvenaud, James Robert Lloyd, Roger B Grosse, Joshua B Tenenbaum, and Zoubin Ghahramani. Structure discovery in nonparametric regression through compositional kernel search. In *International Conference on Machine Learning (3)*, pages 1166–1174, 2013.

[7] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *Uncertainty in Artificial Intelligence*, 2014.

[8] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

[9] S. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. *Neural Information Processing Systems*, 21:793–800, 2008.

[10] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 689–690. IEEE, 2011.

[11] Garvesh Raskutti, Bin Yu, and Martin J Wainwright. Lower bounds on minimax rates for nonparametric regression with additive sparsity and smoothness. In *Advances in Neural Information Processing Systems*, pages 1563–1570, 2009.

[12] Pradeep Ravikumar, Han Liu, John D Lafferty, and Larry A Wasserman. Spam: Sparse additive models. In *Neural Information Processing Systems*, pages 1201–1208, 2007.

[13] Narayana P Santhanam and Martin J Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *IEEE Transactions on Information Theory*, 58(7):4117–4134, 2012.

[14] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

[15] W. Wang, M. Wainwright, and K. Ramchandran. Information-theoretic bounds on model selection for Gaussian Markov random fields. *IEEE International Symposium on Information Theory*, pages 1373 − 1377, 2010.

[16] Bin Yu. Assouad, Fano and Le Cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997.