Learning Sparse Additive Models with Interactions in High Dimensions

Hemant Tyagi
ETH Zürich

Anastasios Kyrillidis
UT Austin, Texas

Bernd Gärtner
ETH Zürich

Andreas Krause
ETH Zürich

Abstract

A function $f : \mathbb{R}^d \to \mathbb{R}$ is referred to as a Sparse Additive Model (SPAM), if it is of the form $f(\mathbf{x}) = \sum_{l \in \mathcal{S}} \phi_l(x_l)$, where $\mathcal{S} \subset [d]$, $|\mathcal{S}| \ll d$. Assuming ϕ_l's and $\mathcal{S}$ to be unknown, the problem of estimating f from its samples has been studied extensively. In this work, we consider a generalized SPAM, allowing for *second order* interaction terms. For some $\mathcal{S}_1 \subset [d], \mathcal{S}_2 \subset \binom{[d]}{2}$, the function f is assumed to be of the form:

$$f(\mathbf{x}) = \sum_{p \in \mathcal{S}_1} \phi_p(x_p) + \sum_{(l,l') \in \mathcal{S}_2} \phi_{(l,l')}(x_l, x_{l'}).$$

Assuming $\phi_p, \phi_{(l,l')}$, $\mathcal{S}_1$ and, $\mathcal{S}_2$ to be unknown, we provide a randomized algorithm that queries f and *exactly recovers* $\mathcal{S}_1, \mathcal{S}_2$. Consequently, this also enables us to estimate the underlying $\phi_p, \phi_{(l,l')}$. We derive sample complexity bounds for our scheme and also extend our analysis to include the situation where the queries are corrupted with noise – either stochastic, or arbitrary but bounded. Lastly, we provide simulation results on synthetic data, that validate our theoretical findings.

1 Introduction

Many scientific problems involve estimating an unknown function f, defined over a compact subset of $\mathbb{R}^d$, with d large. Such problems arise for instance, in modeling complex physical processes [1, 2, 3]. Information about f is typically available in the form of point values $(x_i, f(x_i))_{i=1}^n$, which are then used for learning f. It is well known that the problem suffers from the curse of dimensionality, if only smoothness assumptions are placed on f. For example, if f is C^s smooth, then for uniformly approximating f within error $\delta \in (0,1)$, one needs $n = \Omega(\delta^{-d/s})$ samples [4].

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51.

A popular line of work in recent times considers the setting where f possesses an intrinsic low dimensional structure, *i.e.*, depends on only a small subset of d variables. There exist algorithms for estimating such f (tailored to the underlying structural assumption), along with attractive theoretical guarantees that do not suffer from the curse of dimensionality; see [5, 6, 7, 8]. One such assumption leads to the class of sparse additive models (SPAMs), wherein:

$$f(x_1, \ldots, x_d) = \sum_{l \in \mathcal{S}} \phi_l(x_l),$$

for some unknown $\mathcal{S} \subset \{1, \ldots, d\}$ with $|\mathcal{S}| = k \ll d$. There exist several algorithms for learning these models; we refer to [9, 10, 11, 12, 13] and references therein.

In this paper, we focus on a generalized SPAM model, where f can also contain a small number of *second order interaction terms*, *i.e.*,

$$f(x_1, \ldots, x_d) = \sum_{p \in \mathcal{S}_1} \phi_p(x_p) + \sum_{(l,l') \in \mathcal{S}_2} \phi_{(l,l')}(x_l, x_{l'}); \tag{1.1}$$

$\mathcal{S}_1 \subset [d], \mathcal{S}_2 \subset \binom{[d]}{2}$, with $|\mathcal{S}_1| \ll d, |\mathcal{S}_2| \ll d^2$. There exist relatively few results for learning models of the form (1.1), with the existing work being in the regression framework [14, 15, 16]. Here, $(x_i, f(x_i))_{i=1}^n$ are typically samples from an unknown probability measure $\mathbb{P}$.

We consider the setting where we have the freedom to query f at any desired set of points. We propose a strategy for querying f, along with an efficient recovery algorithm, which leads to much stronger guarantees, compared to those known in the regression setting. In particular, we provide the first *finite sample bounds* for exactly recovering sets $\mathcal{S}_1$ and $\mathcal{S}_2$. Subsequently, we *uniformly* estimate the individual components: $\phi_p, \phi_{(l,l')}$ via additional queries of f along the subspaces corresponding to $\mathcal{S}_1, \mathcal{S}_2$.

Contributions. We make the following contributions for learning models of the form (1.1).

(i) Firstly, we provide a randomized algorithm which provably recovers $\mathcal{S}_1, \mathcal{S}_2$ *exactly*, with $O(k\rho_m(\log d)^3)$ noiseless point queries. Here, ρ_m denotes the maximum number of occurrences of a

variable in $\mathcal{S}_2$, and captures the underlying *complexity* of the interactions.

(ii) An important tool in our analysis is a compressive sensing based sampling scheme, for recovering each row of a sparse Hessian matrix, for functions that also possess sparse gradients. This might be of independent interest.

(iii) We theoretically analyze the impact of additive noise in the point queries on the performance of our algorithm, for two noise models: arbitrary bounded noise and independent, identically distributed (i.i.d.) noise. In particular, for additive Gaussian noise, we show that with $O(\rho_m^5 k^2 (\log d)^4)$ noisy point queries, our algorithm recovers $\mathcal{S}_1, \mathcal{S}_2$ exactly. We also provide simulation results on synthetic data that validate our theoretical findings.

Notation. For any vector $\mathbf{x} \in \mathbb{R}^d$, we denote its ℓ_p-norm by $\|\mathbf{x}\|_p := \left(\sum_{l=1}^d |x_i|^p\right)^{1/p}$. For a set $\mathcal{S}$, $(\mathbf{x})_{\mathcal{S}}$ denotes the restriction of $\mathbf{x}$ onto $\mathcal{S}$, *i.e.*, $((\mathbf{x})_{\mathcal{S}})_l = x_l$ if $l \in \mathcal{S}$ and 0 otherwise. For a function $g : \mathbb{R}^m \to \mathbb{R}$ of m variables, $\mathbb{E}_p[g]$, $\mathbb{E}_{(l,l')}[g], \mathbb{E}[g]$ denote expectation with respect to uniform distributions over $x_p, (x_l, x_{l'})$ and $(x_1, \ldots, x_m)$, respectively. For any compact $\Omega \subset \mathbb{R}^n$, $\|g\|_{L_\infty(\Omega)}$ denotes the L_∞ norm of g in Ω. The partial derivative operator $\partial/\partial x_i$ is denoted by ∂_i. For instance, $\partial_1^2 \partial_2 g$ denotes $\partial^3 g / \partial x_1^2 \partial x_2$.

2 Problem statement

We are interested in the problem of approximating functions $f : \mathbb{R}^d \to \mathbb{R}$ from point queries. For some unknown sets $\mathcal{S}_1 \subset [d], \mathcal{S}_2 \subset \binom{[d]}{2}$, the function f is assumed to have the following form.

$$f(x_1, \ldots, x_d) = \sum_{p \in \mathcal{S}_1} \phi_p(x_p) + \sum_{(l,l') \in \mathcal{S}_2} \phi_{(l,l')}(x_l, x_{l'}). \tag{2.1}$$

Here, $\phi_{(l,l')}$ is considered to be "truly bivariate" meaning that $\partial_l \partial_{l'} \phi_{(l,l')} \not\equiv 0$. The set of all variables that occur in $\mathcal{S}_2$, is denoted by $\mathcal{S}_2^{\text{var}}$. For each $l \in \mathcal{S}_2^{\text{var}}$, we refer to $\rho(l)$ as the *degree* of l, *i.e.*, the number of occurrences of l in $\mathcal{S}_2$, formally defined as:

$$\rho(l) := |\{l' \in \mathcal{S}_2^{\text{var}} : (l,l') \in \mathcal{S}_2 \text{ or } (l',l) \in \mathcal{S}_2\}|; \quad l \in \mathcal{S}_2^{\text{var}}.$$

The largest such degree is denoted by $\rho_m := \max_{l \in \mathcal{S}_2^{\text{var}}} \rho(l)$.

Our goal is to query f at suitably chosen points in its domain, in order to estimate it within the compact region[1] $[-1,1]^d$. To this end, note that representation (2.1) is not unique[2]. This is avoided by re-writing (2.1) in the following unique ANOVA form [17]:

$$f(x_1, \ldots, x_d) = c + \sum_{p \in \mathcal{S}_1} \phi_p(x_p) + \sum_{(l,l') \in \mathcal{S}_2} \phi_{(l,l')}(x_l, x_{l'}) + \sum_{q \in \mathcal{S}_2^{\text{var}} : \rho(q) > 1} \phi_q(x_q), \tag{2.2}$$

where $\mathcal{S}_1 \cap \mathcal{S}_2^{\text{var}} = \emptyset$. Here, $c = \mathbb{E}[f]$ and $\mathbb{E}_p[\phi_p] = \mathbb{E}_{(l,l')}[\phi_{(l,l')}] = 0$; $\forall p \in \mathcal{S}_1, (l,l') \in \mathcal{S}_2$, with expectations being over uniform distributions with respect to variable range $[-1,1]$. In addition, $\mathbb{E}_l[\phi_{(l,l')}] = 0$ if $\rho(l) = 1$. The univariate ϕ_q corresponding to $q \in \mathcal{S}_2^{\text{var}}$ with $\rho(q) > 1$, represents the net marginal effect of the variable and has $\mathbb{E}_q[\phi_q] = 0$. We note that $\mathcal{S}_1, \mathcal{S}_2^{\text{var}}$ are disjoint in (2.2) as each $p \in \mathcal{S}_1 \cap \mathcal{S}_2^{\text{var}}$ can be merged with their bivariate counterparts, uniquely. The uniqueness of (2.2) is shown formally in the appendix.

We assume the setting $|\mathcal{S}_1| = k_1 \ll d$, $|\mathcal{S}_2| = k_2 \ll d^2$. The set of *all* active variables *i.e.*, $\mathcal{S}_1 \cup \mathcal{S}_2^{\text{var}}$ is denoted by $\mathcal{S}$, with $k := |\mathcal{S}| = k_1 + |\mathcal{S}_2^{\text{var}}|$ being the *total sparsity* of the problem.

Due to the special structure of f in (2.2), we note that if $\mathcal{S}_1, \mathcal{S}_2$ were known beforehand, then one can estimate f via standard results from approximation theory or from regression[3]. Hence, our primary focus in the paper is to recover $\mathcal{S}_1, \mathcal{S}_2$. Our main assumptions for this problem are listed below.

Assumption 1. *f can be queried from the slight enlargement: $[-(1+r),(1+r)]^d$, for some small $r > 0$.*

Assumption 2. *Each $\phi_{(l,l')}, \phi_p$ is three times continuously differentiable, within $[-(1+r),(1+r)]^2$ and $[-(1+r),(1+r)]$ respectively. Since these domains are compact, there exist constants $B_m \geq 0$ ($m = 0,1,2,3$) so that:*

$$\|\partial_l^{m_1} \partial_{l'}^{m_2} \phi_{(l,l')}\|_{L_\infty[-(1+r),(1+r)]^2} \leq B_m;\ m_1 + m_2 = m,$$

where $(l,l') \in \mathcal{S}_2$, and

$$\|\partial_p^m \phi_p\|_{L_\infty[-(1+r),(1+r)]} \leq B_m,$$

where $p \in \mathcal{S}_1$ or, $p \in \mathcal{S}_2^{var}$ and $\rho(p) > 1$.

Our next assumption is for identifying $\mathcal{S}_1$.

Assumption 3. *For some constants $D_1, \lambda_1 > 0$, we assume that for each $p \in \mathcal{S}_1$, $\exists$ connected $\mathcal{I}_p \subset [-1,1]$, of Lebesgue measure at least $\lambda_1 > 0$, such that $|\partial_p \phi_p(x_p)| > D_1$, $\forall x_p \in \mathcal{I}_p$. This assumption is in a sense necessary. If say $\partial_p \phi_p$ was zero throughout $[-1,1]$, then it implies that $\phi_p \equiv 0$, since each ϕ_p has zero mean in* (2.2)*.*

[1] One could more generally consider the region $[\alpha,\beta]^d$ and transform the variables to $[-1,1]^d$ via scaling and transformation.

[2] Firstly, we could add constants to each $\phi_l, \phi_{(l,l')}$, which sum up to zero. Furthermore, for each $l \in \mathcal{S}_2^{\text{var}} : \rho(l) > 1$, or $l \in \mathcal{S}_1 \cap \mathcal{S}_2^{\text{var}} : \rho(l) = 1$, we could add univariates that sum to zero.

[3] This is discussed later.

Our last assumption concerns the identification of $\mathcal{S}_2$.

Assumption 4. *For some constants $D_2, \lambda_2 > 0$, we assume that for each $(l,l') \in \mathcal{S}_2$, $\exists$ connected $\mathcal{I}_l, \mathcal{I}_{l'} \subset [-1,1]$, each interval of Lebesgue measure at least $\lambda_2 > 0$, such that $|\partial_l \partial_{l'} \phi_{(l,l')}(x_l, x_{l'})| > D_2$, $\forall (x_l, x_{l'}) \in \mathcal{I}_l \times \mathcal{I}_{l'}$.*

Given the above, our problem specific parameters are: (i) B_i; $i = 0, .., 3$, (ii) D_j, λ_j; $j = 1, 2$ and, (iii) k, ρ_m. We do not assume k_1, k_2 to be known, but instead assume that k is known. Furthermore it suffices to use estimates for the problem parameters instead of exact values: In particular, we can use upper bounds for: k, ρ_m, B_i; $i = 0, .., 3$ and lower bounds for: D_j, λ_j; $j = 1, 2$.

3 Our sampling scheme and algorithm

We start by explaining our sampling scheme, followed by our algorithm for identifying $\mathcal{S}_1, \mathcal{S}_2$. Our algorithm proceeds in two phases – we first estimate $\mathcal{S}_2$ and then $\mathcal{S}_1$. Its theoretical properties for the *noiseless* query setting are described in Section 4. Section 5 then analyzes how the sampling conditions can be adapted to handle the *noisy* query setting.

3.1 Sampling scheme for estimating $\mathcal{S}_2$

Our main idea for estimating $\mathcal{S}_2$ is to estimate the off-diagonal entries of the Hessian of f, at appropriately chosen points. The motivation is the observation that for any $(l,l') \in \binom{[d]}{2}$:

$$\partial_l \partial_{l'} f = \begin{cases} \partial_l \partial_{l'} \phi_{(l,l')} & \text{if } (l,l') \in \mathcal{S}_2, \\ 0 & \text{otherwise.} \end{cases}$$

To this end, consider the Taylor expansion of the gradient ∇f, at $\mathbf{x} \in \mathbb{R}^d$, along the direction $\mathbf{v}' \in \mathbb{R}^d$, with step size μ_1. Since f is C^3 smooth, we have for $\zeta_q = \mathbf{x} + \theta_q \mathbf{v}'$, for some $\theta_q \in (0, \mu_1)$, $q = 1, \ldots, d$:

$$\frac{\nabla f(\mathbf{x} + \mu_1 \mathbf{v}') - \nabla f(\mathbf{x})}{\mu_1} = \nabla^2 f(\mathbf{x}) \mathbf{v}' + \frac{\mu_1}{2} \begin{pmatrix} \mathbf{v}'^T \nabla^2 \partial_1 f(\zeta_1) \mathbf{v}' \\ \vdots \\ \mathbf{v}'^T \nabla^2 \partial_d f(\zeta_d) \mathbf{v}' \end{pmatrix}. \quad (3.1)$$

We see from (3.1) that the l^{th} entry of $(\nabla f(\mathbf{x} + \mu_1 \mathbf{v}') - \nabla f(\mathbf{x}))/\mu_1$, corresponds to a "noisy" linear measurement of the l^{th} row of $\nabla^2 f(\mathbf{x})$ with $\mathbf{v}'$. The noise corresponds to the third order Taylor remainder terms of f.

Denoting the l^{th} row of $\nabla^2 f(\mathbf{x})$ by $\nabla \partial_l f(\mathbf{x}) \in \mathbb{R}^d$, we make the following crucial observation: if $l \in \mathcal{S}_2^{\text{var}}$ then $\nabla \partial_l f(\mathbf{x})$ has at most ρ_m non-zero *off-diagonal* entries, implying that it is $(\rho_m + 1)$ sparse. This follows on account of the structure of f (2.2). Furthermore, if $l \in \mathcal{S}_1$ then $\nabla \partial_l f(\mathbf{x})$ has at most one non zero entry (namely the diagonal entry), while if $l \notin \mathcal{S}$, then $\nabla \partial_l f(\mathbf{x}) \equiv 0$.

Compressive sensing based estimation. Assuming for now that we have access to an oracle that provides us with gradient estimates of f, this suggests the following idea. We can obtain random linear measurements, for *each row* of $\nabla^2 f(\mathbf{x})$ via gradient differences, as in (3.1). As each row is sparse, it is known from compressive sensing (CS) [18, 19] that it can be recovered with only a few measurements.

Inspired by this observation, consider an oracle that provides us with the estimates: $\widehat{\nabla} f(\mathbf{x}), \{\widehat{\nabla} f(\mathbf{x} + \mu_1 \mathbf{v}'_j)\}_{j=1}^{m_{v'}}$ where $\mathbf{v}'_j$ belong to the set:

$$\mathcal{V}' := \{\mathbf{v}'_j \in \mathbb{R}^d : v'_{j,q} = \pm 1/\sqrt{m_{v'}} \text{ w.p. } 1/2 \text{ each}; \; j = 1, \ldots, m_{v'} \text{ and } q = 1, \ldots, d\}.$$

Let $\widehat{\nabla} f(\mathbf{x}) = \nabla f(\mathbf{x}) + \mathbf{w}(\mathbf{x})$, where $\mathbf{w}(\mathbf{x}) \in \mathbb{R}^d$ denotes the gradient estimation noise. Denoting $\mathbf{V}' = [\mathbf{v}'_1 \ldots \mathbf{v}'_{m_{v'}}]^T$, we obtain d linear systems, by employing (3.1) at each $\mathbf{v}'_j \in \mathcal{V}'$:

$$\mathbf{y}_q = \mathbf{V}' \nabla \partial_q f(\mathbf{x}) + \eta_{\mathbf{q},\mathbf{1}} + \eta_{\mathbf{q},\mathbf{2}}; \quad q = 1, \ldots, d. \quad (3.2)$$

$\mathbf{y}_q \in \mathbb{R}^{m_{v'}}$ represents the measurement vector for the q^{th} row, with

$$(\mathbf{y}_q)_j = ((\widehat{\nabla} f(\mathbf{x} + \mu_1 \mathbf{v}'_j) - \widehat{\nabla} f(\mathbf{x}))_q)/\mu_1$$

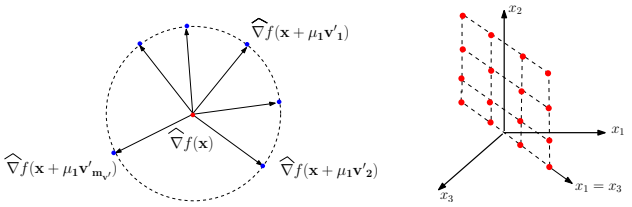
while $\eta_{\mathbf{q},\mathbf{1}}, \eta_{\mathbf{q},\mathbf{2}} \in \mathbb{R}^{m_{v'}}$ represent noise with $(\eta_{\mathbf{q},\mathbf{1}})_j = (\mu_1/2) \mathbf{v}'^T_j \nabla^2 \partial_q f(\zeta_{q,j}) \mathbf{v}'_j$ and $(\eta_{\mathbf{q},\mathbf{2}})_j = (w_q(\mathbf{x} + \mu_1 \mathbf{v}'_j) - w_q(\mathbf{x}))/\mu_1$. Given the measurement vector $\mathbf{y}_q$, we can then obtain the estimate $\widehat{\nabla} \partial_q f(\mathbf{x})$ individually for each $q = 1, \ldots, d$, via ℓ_1 minimization [18, 19, 20].

Estimating sufficiently many Hessian's. Having estimated *each row* of $\nabla^2 f$ at some fixed $\mathbf{x}$, we have at hand an estimate of the set: $\{\partial_i \partial_j f(\mathbf{x}) : (i,j) \in \binom{[d]}{2}\}$. Our next goal is to repeat the process, at sufficiently many $\mathbf{x}$'s within $[-1,1]^d$.

We will denote the set of such points as χ. This will then enable us to sample each underlying $\partial_l \partial_{l'} \phi_{(l,l')}$ within its respective critical interval, as defined in Assumption 4. Roughly speaking, since $|\partial_l \partial_{l'} \phi_{(l,l')}|$ is "suitably large" in such an interval, we will consequently be able to detect each $(l,l') \in \mathcal{S}_2$, via a thresholding procedure. To this end, we make use of a family of hash functions, defined as follows.

Definition 1. *For some $t \in \mathbb{N}$ and $j = 1, 2, \ldots,$ let $h_j : [d] \to \{1, 2, \ldots, t\}$. Then, the set $\mathcal{H}_t^d = \{h_1, h_2, \ldots\}$ is a (d,t)-hash family if for any distinct $i_1, \ldots, i_t \in [d]$, $\exists$ $h \in \mathcal{H}_t^d$ such that h is an injection when restricted to $i_1, i_2, \ldots, i_t$.*

Hash functions are widely used in theoretical computer science, such as in finding juntas [21]. There exists a simple probabilistic method for constructing such a family, so that



(a) (b)

Figure 1: (a) $\nabla^2 f(\mathbf{x})$ estimated using: $\widehat{\nabla} f(\mathbf{x})$ (at red disk) and neighborhood gradient estimates (at blue disks) (b) Geometric picture: $d = 3$, $h \in \mathcal{H}_2^3$ with $h(1) = h(3) \neq h(2)$. Red disks are points in $\chi(h)$.

for any constant $C > 1$, $|\mathcal{H}_t^d| \leq (C+1)te^t \log d$ with high probability (w.h.p)[4] [5]. For our purposes, we consider the family $\mathcal{H}_2^d$ so that for any distinct i, j, there exists $h \in \mathcal{H}_2^d$ such that $h(i) \neq h(j)$.

For any $h \in \mathcal{H}_2^d$, let us now denote the vectors $\mathbf{e}_1(h), \mathbf{e}_2(h) \in \mathbb{R}^d$ where

$$(\mathbf{e}_i(h))_q = \begin{cases} 1 & \text{if } h(q) = i, \\ 0 & \text{otherwise,} \end{cases}$$

for $i = 1, 2$ and $q = 1, \ldots, d$. Given at hand $\mathcal{H}_2^d$, we construct our set χ using the procedure[5] in [5]. Specifically, for some $m_x \in \mathbb{Z}^+$, we construct for each $h \in \mathcal{H}_2^d$ the set:

$$\chi(h) := \Bigg\{ \mathbf{x}(h) \in [-1,1]^d : \mathbf{x}(h) = \sum_{i=1}^{2} c_i \mathbf{e}_i(h); \\ c_1, c_2 \in \left\{ -1, -\frac{m_x - 1}{m_x}, \ldots, \frac{m_x - 1}{m_x}, 1 \right\} \Bigg\}.$$

Then, we obtain $\chi = \cup_{h \in \mathcal{H}_2^d} \chi(h)$ as the set of points at which we will recover $\nabla^2 f$. Observe that χ has the property of discretizing *any* 2-dimensional canonical subspace, within $[-1,1]^d$ with $|\chi| \leq (2m_x + 1)^2 |\mathcal{H}_2^d| = O(\log d)$.

Estimating sparse gradients. Note that ∇f is at most k sparse, due to the structure of f. We now describe the oracle that we use, for estimating sparse gradients. As f is $\mathcal{C}^3$ smooth, therefore the third order Taylor's expansion of f at $\mathbf{x}$, along $\mathbf{v}, -\mathbf{v} \in \mathbb{R}^d$, with step size $\mu > 0$, and $\zeta = \mathbf{x} + \theta\mathbf{v}$, $\zeta' = \mathbf{x} - \theta'\mathbf{v}$; $\theta, \theta' \in (0, \mu)$ leads to

$$\frac{f(\mathbf{x}+\mu\mathbf{v}) - f(\mathbf{x}-\mu\mathbf{v})}{2\mu} \\ = \langle \mathbf{v}, \nabla f(\mathbf{x}) \rangle + (R_3(\zeta) - R_3(\zeta'))/(2\mu). \tag{3.3}$$

(3.3) corresponds to a noisy-linear measurement of $\nabla f(\mathbf{x})$, with $\mathbf{v}$. The "noise" here arises on account of the third order terms $R_3(\zeta), R_3(\zeta') = O(\mu^3)$, in the Taylor expansion.

[4]With probability $1 - O(d^{-c})$ for some constant $c > 0$.

[5]Such sets were used in [5] for a more general problem involving functions that are intrinsically k variate.

Let $\mathcal{V}$ denote the set of measurement vectors:

$$\mathcal{V} := \{ v_j \in \mathbb{R}^d : v_{j,q} = \pm 1/\sqrt{m_v} \text{ w.p. } 1/2 \text{ each}; \\ j = 1, \ldots, m_v \text{ and } q = 1, \ldots, d \}.$$

Employing (3.3) at each $\mathbf{v}_j \in \mathcal{V}$, we obtain:

$$\mathbf{y} = \mathbf{V} \nabla f(\mathbf{x}) + \mathbf{n}. \tag{3.4}$$

Here, $\mathbf{y} \in \mathbb{R}^{m_v}$ denotes the measurement vector with $(\mathbf{y})_j = (f(\mathbf{x} + \mu\mathbf{v}_j) - f(\mathbf{x} - \mu\mathbf{v}_j))/(2\mu)$. Also, $\mathbf{V} = [\mathbf{v}_1 \ldots \mathbf{v}_{m_v}]^T \in \mathbb{R}^{m_v \times d}$ denotes the measurement matrix and $\mathbf{n} \in \mathbb{R}^{m_v}$ denotes the noise terms. We then estimate $\nabla f(\mathbf{x})$ via standard ℓ_1 minimization[6] [18, 19, 20]. Estimating sparse gradients via CS, has been considered previously in [8, 13], albeit using *second order* Taylor expansions, for different function models.

3.2 Sampling scheme for estimating $\mathcal{S}_1$

Having obtained an estimate $\widehat{\mathcal{S}_2}$ of $\mathcal{S}_2$ we now proceed to estimate $\mathcal{S}_1$. Let $\widehat{\mathcal{S}_2^{\text{var}}}$ denote the set of variables in $\widehat{\mathcal{S}_2}$ and $\mathcal{P} := [d] \setminus \widehat{\mathcal{S}_2^{\text{var}}}$. Assuming $\widehat{\mathcal{S}_2} = \mathcal{S}_2$, we are now left with a SPAM on the *reduced* variable set $\mathcal{P}$. Consequently, we employ the sampling scheme of [13], wherein the gradient of f is estimated at equispaced points, along a diagonal of $[-1,1]^d$. For $m'_x \in \mathbb{Z}^+$, this set is defined as:

$$\chi_{\text{diag}} := \Bigg\{ \mathbf{x} = (x \; x \; \cdots \; x) \in \mathbb{R}^d : \\ x \in \left\{ -1, -\frac{m'_x - 1}{m'_x}, \ldots, \frac{m'_x - 1}{m'_x}, 1 \right\} \Bigg\}.$$

Note that $|\chi_{\text{diag}}| = 2m'_x + 1$. The motivation for estimating ∇f at $\mathbf{x} \in \chi_{\text{diag}}$ is that we obtain estimates of $\partial_p \phi_p$ at equispaced points within $[-1,1]$, for $p \in \mathcal{S}_1$. With a sufficiently fine discretization, we would "hit" the critical regions associated with each $\partial_p \phi_p$, as defined in Assumption 3. By applying a thresholding operation, we would then be able to identify each $p \in \mathcal{S}_1$.

To this end, consider the set of sampling directions:

$$\mathcal{V}'' := \{ \mathbf{v}''_j \in \mathbb{R}^d : v''_{j,q} = \pm 1/\sqrt{m_{v''}} \text{ w.p. } 1/2 \text{ each}; \\ j = 1, \ldots, m_{v''} \text{ and } q = 1, \ldots, d \},$$

and let $\mu' > 0$ denote the step size. For each $\mathbf{x} \in \chi_{\text{diag}}$, we will query f at points: $(\mathbf{x} + \mu'\mathbf{v}''_j)_{\mathcal{P}}, (\mathbf{x} - \mu'\mathbf{v}''_j)_{\mathcal{P}}$; $\mathbf{v}''_j \in \mathcal{V}''$, *restricted* to $\mathcal{P}$. Then, as described earlier, we can form a linear system consisting of $m_{v''}$ equations, and solve it via ℓ_1 minimization to obtain the gradient estimate. The complete procedure for estimating $\mathcal{S}_1, \mathcal{S}_2$, is described formally in Algorithm 1.

[6]Can be solved efficiently using interior point methods [22]

Algorithm 1 Algorithm for estimating $\mathcal{S}_1, \mathcal{S}_2$

Input: $m_v, m_{v'}, m_x, m'_x \in \mathbb{Z}^+$; $\mu, \mu_1, \mu' > 0$; $\tau' > 0, \tau'' > 0$.
Initialization: $\widehat{\mathcal{S}_1}, \widehat{\mathcal{S}_2} = \emptyset$.
Output: Estimates $\widehat{\mathcal{S}_2}$, $\widehat{\mathcal{S}_1}$.

Construct $(d, 2)$-hash family $\mathcal{H}_2^d$ and sets $\mathcal{V}, \mathcal{V}'$.
for $h \in \mathcal{H}_2^d$ **do**
  Construct the set $\chi(h)$.
  for $i = 1, \ldots, (2m_x+1)^2$ and $\mathbf{x}_i \in \chi(h)$ **do**
    $(\mathbf{y}_i)_j = \frac{f(\mathbf{x}_i+\mu\mathbf{v}_j)-f(\mathbf{x}_i-\mu\mathbf{v}_j)}{2\mu}$; $j = 1, \ldots, m_v$; $\mathbf{v}_j \in \mathcal{V}$.
    $\widehat{\nabla} f(\mathbf{x}_i) := \underset{\mathbf{y}_i=\mathbf{V}\mathbf{z}}{\operatorname{argmin}} \|\mathbf{z}\|_1$.
    for $p = 1, \ldots, m_{v'}$ **do**
      $(\mathbf{y}_{i,p})_j = \frac{f(\mathbf{x}_i+\mu_1\mathbf{v}'_p+\mu\mathbf{v}_j)-f(\mathbf{x}_i+\mu_1\mathbf{v}'_p-\mu\mathbf{v}_j)}{2\mu}$; $j = 1, \ldots, m_v$; $\mathbf{v}'_p \in \mathcal{V}'$. ESTIMATION OF $\mathcal{S}_2$
      $\widehat{\nabla} f(\mathbf{x}_i + \mu_1\mathbf{v}'_p) := \underset{\mathbf{y}_{i,p}=\mathbf{V}\mathbf{z}}{\operatorname{argmin}} \|\mathbf{z}\|_1$.
    end for
    for $q = 1, \ldots, d$ **do**
      $(\mathbf{y}_q)_j = \frac{(\widehat{\nabla} f(\mathbf{x}_i+\mu_1\mathbf{v}'_j)-\widehat{\nabla} f(\mathbf{x}_i))_q}{\mu_1}$; $j = 1, \ldots, m_{v'}$.
      $\widehat{\nabla}\partial_q f(\mathbf{x}_i) := \underset{\mathbf{y}_q=\mathbf{V}'\mathbf{z}}{\operatorname{argmin}} \|\mathbf{z}\|_1$.
      $\widehat{\mathcal{S}_2} = \widehat{\mathcal{S}_2} \cup \left\{(q, q') : q' \in \{q+1, \ldots, d\} \ \& \ |(\widehat{\nabla}\partial_q f(\mathbf{x}_i))_{q'}| > \tau'\right\}$.
    end for
  end for
end for

Construct the sets $\chi_{\text{diag}}, \mathcal{V}''$ and initialize $\mathcal{P} := [d] \setminus \widehat{\mathcal{S}_2^{\text{var}}}$.
for $i = 1, \ldots, (2m'_x+1)$ and $\mathbf{x}_i \in \chi_{\text{diag}}$ **do**
  $(\mathbf{y}_i)_j = \frac{f((\mathbf{x}_i+\mu'\mathbf{v}''_j)_\mathcal{P})-f((\mathbf{x}_i-\mu'\mathbf{v}''_j)_\mathcal{P})}{2\mu'}$; $j = 1, \ldots, m_{v''}$; $\mathbf{v}_j \in \mathcal{V}''$.
  $(\widehat{\nabla} f((\mathbf{x}_i)_\mathcal{P}))_\mathcal{P} := \underset{\mathbf{y}_i=(\mathbf{V}'')_\mathcal{P}(\mathbf{z})_\mathcal{P}}{\operatorname{argmin}} \|(\mathbf{z})_\mathcal{P}\|_1$. ESTIMATION OF $\mathcal{S}_1$
  $\widehat{\mathcal{S}_1} = \widehat{\mathcal{S}_1} \cup \left\{q \in \mathcal{P} : |((\widehat{\nabla} f((\mathbf{x}_i)_\mathcal{P})_q| > \tau''\right\}$.
end for

4 Theoretical guarantees for noiseless case

Next, we provide sufficient conditions on our sampling parameters that guarantee exact recovery of $\mathcal{S}_1, \mathcal{S}_2$, in the noiseless query setting. This is stated in the following Theorem. All proofs are deferred to the appendix.

Theorem 1. *$\exists$ positive constants $\{c'_i\}_{i=1}^3, \{C_i\}_{i=1}^3$ so that if: $m_x \geq \lambda_2^{-1}$, $m_v > c'_1 k \log(d/k)$, and $m_{v'} > c'_2\rho_m \log(d/\rho_m)$, then the following holds. Denoting $a = \frac{(4\rho_m+1)B_3}{2\sqrt{m_{v'}}}$, $b = \frac{C_1\sqrt{m_{v'}}((4\rho_m+1)k)B_3}{3m_v}$, $a' = \frac{D_2}{4aC_2}$, let μ, μ_1 satisfy: $\mu^2 < (a'^2 a)/b$ and*

$$\mu_1 \in (a' - \sqrt{a'^2 - (b\mu^2/a)}, a' + \sqrt{a'^2 - (b\mu^2/a)}).$$

We then have for $\tau' = C_2(a\mu_1 + \frac{b\mu^2}{\mu_1})$, that $\widehat{\mathcal{S}_2} = \mathcal{S}_2$ w.h.p. Provided $\widehat{\mathcal{S}_2} = \mathcal{S}_2$, if $m'_x \geq \lambda_1^{-1}$, $m_{v''} > c'_3(k - |\widehat{\mathcal{S}_2^{var}}|)\log(\frac{|\mathcal{P}|}{k-|\widehat{\mathcal{S}_2^{var}}|})$ and $\mu'^2 < \frac{3m_{v''}D_1}{C_3(k-|\widehat{\mathcal{S}_2^{var}}|)B_3}$, then $\tau'' = \frac{C_3(k-|\widehat{\mathcal{S}_2^{var}}|)\mu'^2 B_3}{6m_{v''}}$, implies $\widehat{\mathcal{S}_1} = \mathcal{S}_1$ w.h.p.

Remark 1. *We note that the condition on μ' is less strict than in [13] for identifying $\mathcal{S}_1$. This is because in [13], the gradient is estimated via a forward difference procedure, while we perform a central difference procedure in* (3.3).

Query complexity. Estimating $\nabla f(\mathbf{x})$ at some fixed $\mathbf{x}$ requires $2m_v = O(k \log d)$ queries. Estimating $\nabla^2 f(\mathbf{x})$ involves computing an additional $m_{v'} = O(\rho_m \log d)$ gradient vectors in a neighborhood of $\mathbf{x}$ – implying $O(m_v m_{v'}) = O(k\rho_m(\log d)^2)$ point queries. This consequently implies a total query complexity of $O(k\rho_m(\log d)^2|\chi|) = O(\lambda_2^{-2}k\rho_m(\log d)^3)$, for estimating $\mathcal{S}_2$. We make an additional $O(\lambda_1^{-1}(k - |\widehat{\mathcal{S}_2^{\text{var}}}|)\log(d - |\widehat{\mathcal{S}_2^{\text{var}}}|))$ queries of f, in order to estimate $\mathcal{S}_1$. Therefore, the overall query complexity for estimating $\mathcal{S}_1, \mathcal{S}_2$ is $O(\lambda_2^{-2}k\rho_m(\log d)^3)$.

$\mathcal{H}_2^d$ can be constructed in poly(d) time. For each $\mathbf{x} \in \chi$, we first solve $m_{v'}+1$ linear programs (Steps 10, 13), each solvable in poly(m_v, d) time. We then solve d linear programs (Step 17), with each taking poly$(m_{v'}, d)$ time. This is done at $|\chi| = O(\lambda_2^{-2}\log d)$ points, hence the overall *computation cost* for estimation of $\mathcal{S}_2$ (and later $\mathcal{S}_1$) is polynomial in: the number of queries, and d. Lastly, we note that [23] also estimates sparse Hessians via CS, albeit for the function optimization problem. Their scheme entails a sample complexity[7] of $O(k\rho_m(\log(k\rho_m))^2(\log d)^2)$ for estimating $\nabla^2 f(\mathbf{x})$; this is worse by a $O((\log(k\rho_m))^2)$ term compared to our method.

Recovering the components of the model. Having estimated $\mathcal{S}_1, \mathcal{S}_2$, we can now estimate each underlying component in (2.2) by sampling f along the *subspace* corresponding to the component. Using these samples, one can then construct via standard techniques, a spline based quasi interpolant [24] that *uniformly* approximates the component. This is shown formally in the appendix.

5 Impact of noise

We now consider the case where the point queries are corrupted with external noise. This means that at query $\mathbf{x}$, we observe $f(\mathbf{x})+z'$, where $z' \in \mathbb{R}$ denotes external noise.

In order to estimate $\nabla f(\mathbf{x})$, we obtain the samples : $f(\mathbf{x}+\mu\mathbf{v}_j)+z'_{j,1}$ and $f(\mathbf{x}-\mu\mathbf{v}_j)+z'_{j,2}$; $j=1,\ldots,m_v$. This changes (3.4) to the linear system $\mathbf{y} = \mathbf{V}\nabla f(\mathbf{x}) + \mathbf{n} + \mathbf{z}$, where $z_j = (z'_{j,1}-z'_{j,2})/(2\mu)$. Hence, the step-size μ needs to be chosen carefully now – a small value would blow up the external noise component, while a large value would increase perturbation due to the higher order Taylor's terms.

Arbitrary bounded noise. In this scenario, we assume the external noise to be arbitrary and bounded, meaning that $|z'| < \varepsilon$, for some finite $\varepsilon \geq 0$. If ε is too large, then we would expect recovery of $\mathcal{S}_1, \mathcal{S}_2$ to be impossible as the structure of f would be destroyed.

We show in Theorem 2 that if $\varepsilon < \varepsilon_1 = O\left(D_2^3/(B_3^2\rho_m^2\sqrt{k})\right)$, then Algorithm 1 recovers $\mathcal{S}_2$ with appropriate choice of sampling parameters. Furthermore, assuming $\mathcal{S}_2$ is recovered exactly, and provided ε additionally satisfies $\varepsilon < \varepsilon_2 = O\left(D_1^{3/2}/\sqrt{(k-|\widehat{\mathcal{S}_2^{\text{var}}}|)B_3}\right)$, then the algorithm also recovers $\mathcal{S}_1$ exactly. In contrast to Theorem 1, the step size μ cannot be chosen arbitrarily small now, due to external noise.

Theorem 2. *Let $m_x, m'_x, m_v, m_{v'}, m_{v''}$ be as defined in Theorem 1. Say $\varepsilon < \varepsilon_1 = O\left(\frac{D_2^3}{B_3^2\rho_m^2\sqrt{k}}\right)$. Denoting $b' = 2C_1\sqrt{m_v m_{v'}}$, $\exists 0 < A_1 < A_2$ and $0 < A_3 < A_4$ so that for $\mu \in (A_1, A_2)$, $\mu_1 \in (A_3, A_4)$ and $\tau' = C_2(a\mu_1 + \frac{b\mu^2}{\mu_1} + \frac{b'\varepsilon}{\mu\mu_1})$, we have $\widehat{\mathcal{S}_2} = \mathcal{S}_2$ w.h.p. Given $\widehat{\mathcal{S}_2} = \mathcal{S}_2$, denote $a_1 = (k-|\widehat{\mathcal{S}_2^{\text{var}}}|)B_3/(6m_{v''})$, $b_1 = \sqrt{m_{v''}}$ and say $\varepsilon < \varepsilon_2 = O\left(\frac{D_1^{3/2}}{\sqrt{(k-|\widehat{\mathcal{S}_2^{\text{var}}}|)B_3}}\right)$. $\exists 0 < A_5 < A_6$ so that $\mu' \in (A_5, A_6)$, $\tau'' = C_3(a_1\mu'^2 + \frac{b_1\varepsilon}{\mu'})$ implies $\widehat{\mathcal{S}_1} = \mathcal{S}_1$ w.h.p.*

Stochastic noise. We now assume the point queries to be corrupted with i.i.d. Gaussian noise, so that $z' \sim \mathcal{N}(0, \sigma^2)$ with variance σ^2. We consider resampling each point query a sufficient number of times, and averaging the values. During the $\mathcal{S}_2$ estimation phase, we resample each query N_1 times so that $z' \sim \mathcal{N}(0, \sigma^2/N_1)$. For any $0 < \varepsilon < \varepsilon_1$, if N_1 is suitably large, then we can uniformly bound $|z'| < \varepsilon$ – via standard tail bounds for Gaussians – over all noise samples, with high probability. Consequently, we can use the result of Theorem 2 for estimating $\mathcal{S}_2$. The same reasoning applies to Step 25, *i.e.*, the $\mathcal{S}_1$ estimation phase, where we resample each query N_2 times.

Theorem 3. *Let $m_x, m'_x, m_v, m_{v'}, m_{v''}$ be as defined in Theorem 1. For any $\varepsilon < \varepsilon_1$, $0 < p_1 < 1$, say we resample each query in Steps 9, 12, $N_1 > \frac{\sigma^2}{\varepsilon^2}\log(\frac{\sqrt{2}\sigma}{\varepsilon p_1}m_v(m_{v'}+1)(2m_x+1)^2|\mathcal{H}_2^d|)$ times, and take the average. For μ, μ_1, τ' as in Theorem 2, we have $\widehat{\mathcal{S}_2} = \mathcal{S}_2$ with probability at least $1-p_1-o(1)$. Given $\widehat{\mathcal{S}_2} = \mathcal{S}_2$, with $\varepsilon' < \varepsilon_2$, $0 < p_2 < 1$, say we resample each query in Step 25, $N_2 > \frac{\sigma^2}{\varepsilon'^2}\log(\frac{\sqrt{2}\sigma(2m'_x+1)m_{v''}}{\varepsilon' p_2})$ times, and take the average. Then for μ', τ'' as in Theorem 2 (with ε replaced by ε'), we have $\widehat{\mathcal{S}_1} = \mathcal{S}_1$ with probability at least $1-p_2-o(1)$.*

Query complexity. In the case of arbitrary, but bounded noise, the query complexity remains the same as for the noiseless case. In case of i.i.d. Gaussian noise, for estimating $\mathcal{S}_2$, we have $\varepsilon = O(\rho_m^{-2}k^{-1/2})$. Choosing $p_1 = d^{-\delta}$ for any constant $\delta > 0$ gives us $N_1 = O(\rho_m^4 k\log d)$. This means that with $O(N_1 k\rho_m(\log d)^3|\chi|) = O(\rho_m^5 k^2(\log d)^4\lambda_2^{-2})$ queries, $\widehat{\mathcal{S}_2} = \mathcal{S}_2$ holds w.h.p. Next, for estimating $\mathcal{S}_1$, we have $\varepsilon' = O((k-|\mathcal{S}_2^{\text{var}}|)^{-1/2})$. Choosing $p_2 = ((d-|\mathcal{S}_2^{\text{var}}|)^{-\delta})$ for any constant $\delta > 0$, we get $N_2 = O((k-|\mathcal{S}_2^{\text{var}}|)\log(d-|\mathcal{S}_2^{\text{var}}|))$. This means the query complexity for estimating $\mathcal{S}_1$ is $O(N_2\lambda_1^{-1}(k-|\widehat{\mathcal{S}_2^{\text{var}}}|)\log(d-|\widehat{\mathcal{S}_2^{\text{var}}}|)) = O(\lambda_1^{-1}(k-|\widehat{\mathcal{S}_2^{\text{var}}}|)^2(\log(d-|\widehat{\mathcal{S}_2^{\text{var}}}|))^2)$. Therefore, the overall query complexity of Algorithm 1 for estimating $\mathcal{S}_1, \mathcal{S}_2$ is $O(\rho_m^5 k^2(\log d)^4\lambda_2^{-2})$.

Remark 2. *We saw above that $O(k^2(\log d)^2)$ samples are sufficient for estimating $\mathcal{S}_1$ in presence of i.i.d Gaussian noise. This improves the corresponding bound in [13] by a $O(k)$ factor, and is due to the less strict condition on μ'.*

Recovering the components of the model. Having identified $\mathcal{S}_1, \mathcal{S}_2$, we can estimate the underlying components in (2.2), via standard nonparametric regression for ANOVA type models [25]. Alternately, for each component, we could also sample f along the subspace corresponding to

[7] See [23, Corollary 4.1]

the component and then perform regression, to obtain its estimate with *uniform* error bounds. This is shown formally in the appendix.

6 Related work

Learning SPAMs. We begin with an overview of results for learning SPAMs, in the regression setting. [14] proposed the COSSO algorithm, that extends the Lasso to the reproducing kernel Hilbert space (RKHS) setting. [26] generalizes the non negative garrote to the nonparametric setting. [27, 9, 10] consider least squares methods, regularized by sparsity inducing penalty terms, for learning such models. [12, 28] propose a convex program for estimating f (in the RKHS setting) that achieves the minimax optimal error rates. [11] proposes a method based on the adaptive group Lasso. These methods are designed for learning SPAMs and cannot handle models of the form (1.1).

Learning generalized SPAMs. There exist fewer results for generalized SPAMs of the form (1.1), in the regression setting. The COSSO algorithm [14] can handle (1.1), however its convergence rates are shown only for the case of no interactions. [15] proposes the VANISH algorithm – a least squares method with sparsity constraints. It is shown to be sparsistent, *i.e.*, it asymptotically recovers $\mathcal{S}_1, \mathcal{S}_2$ for $n \to \infty$. They also show a consistency result for estimating f, similar to [9]. [16] proposes the ACOSSO method, an adaptive version of the COSSO algorithm, which can also handle (1.1). They derive convergence rates and sparsistency results for their method, albeit for the case of no interactions. [29] studies a generalization of (1.1) that allows for the presence of a sparse number of m-wise interaction terms for some additional sparsity parameter m. While they derive[8] non-asymptotic L_2 error rates for estimating f, they do not guarantee unique identification of the interaction terms for any value of m. A special case of (1.1) – where ϕ_p's are linear and each $\phi_{(l,l')}$ is of the form $x_l x_{l'}$ – has been studied considerably. Within this setting, there exist algorithms that recover $\mathcal{S}_1, \mathcal{S}_2$, along with convergence rates for estimating f, but only in the limit of large n [30, 15, 31]. [32] generalized this to the setting of sparse multilinear systems – albeit in the noiseless setting – and derived non-asymptotic sampling bounds for identifying the interaction terms. However finite sample bounds for the non-linear model (1.1) are not known in general.

Learning generic low-dimensional function models. There exists related work in approximation theory – which is also the setting considered in this paper – wherein one assumes freedom to query f at any desired set of points within its domain. [5] considers functions depending on an unknown subset $\mathcal{S}$ ($|\mathcal{S}| = k$) of the variables – a more general model than (1.1). They provide a choice of query points of size $O(c^k k \log d)$ for some constant $c > 1$, and algorithms that recover $\mathcal{S}$ w.h.p. [33] derives a simpler algorithm with sample complexity $O((C_1^4/\alpha^4)k(\log d)^2)$ for recovering $\mathcal{S}$ w.h.p., where C_1, α depend[9] on smoothness of f. For general k-variate f: $\alpha = c^{-k}$ for some constant $c > 1$, while for our model (1.1): $C_1 = O(\rho_m)$. This model was also studied in [34, 35] in the regression setting – they proposed an estimator that recovers $\mathcal{S}$ w.h.p, with sample complexity $O(c^k k \log d)$. [8, 7] generalize this model to functions f of the form $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$, for unknown $\mathbf{A} \in \mathbb{R}^{k \times d}$. They derive algorithms that approximately recover the row-span of $\mathbf{A}$ w.h.p, with sample complexities typically polynomial in d.

While the above methods could possibly recover $\mathcal{S}$, they are not designed for identifying *interactions* among the variables. Specifically, their sample complexities exhibit a worse dependence on k, ρ_m and/or d. [13] provides a sampling scheme that specifically learns SPAMs, with sample complexities $O(k \log d)$, $O(k^3(\log d)^2)$, in the absence/presence of Gaussian noise, respectively.

7 Simulation results

Dependence on d. We first consider the following experimental setup: $\mathcal{S}_1 = \{1, 2\}$ and $\mathcal{S}_2 = \{(3, 4), (4, 5)\}$, which implies $k_1 = 2$, $k_2 = 2$, $\rho_m = 2$ and $k = 5$. We consider two models:

(i) $f_1(\mathbf{x}) = 2x_1 - 3x_2^2 + 4x_3x_4 - 5x_4x_5$,

(ii) $f_2(\mathbf{x}) = 10\sin(\pi \cdot x_1) + 5e^{-2x_2} + 10\sin(\pi \cdot x_3x_4) + 5e^{-2x_4x_5}$.

We begin with the relatively simple model f_1, for which the problem parameters are set to: $\lambda_1 = 0.3$, $\lambda_2 = 1$, $D_1 = 2$, $D_2 = 3$, $B_3 = 6$. We obtain $m_x = 1$, $m_x' = 4$. We use the same constant $\widetilde{C}$ when we set $m_v := \widetilde{C}k\log(d/k)$, $m_{v'} := \widetilde{C}\rho_m \log(d/\rho_m)$, and $m_{v''} := \widetilde{C}(k - |\widehat{\mathcal{S}_2^{\text{var}}}|)\log(\frac{|\mathcal{P}|}{k-|\widehat{\mathcal{S}_2^{\text{var}}}|})$. For the construction of the hash functions, we set the size to $|\mathcal{H}_2^d| = C' \log d$ with $C' = 1.7$, leading to $|\mathcal{H}_2^d| \in [8, 12]$ for $10^2 \leq d \leq 10^3$. We choose step sizes: μ, μ_1, μ' and thresholds: τ', τ'' as in Theorem 2. As CS solver, we use the ALPS algorithm [36], an efficient first-order method.

For the noisy setting, we consider the function values to be corrupted with i.i.d. Gaussian noise. The noise variance values considered are: $\sigma^2 \in \{10^{-4}, 10^{-3}, 10^{-2}\}$ for which we choose resampling factors: $(N_1, N_2) \in \{(50, 20), (85, 36), (90, 40)\}$. We see in Fig. 2, that for $\widetilde{C} \approx 5.6$ the probability of successful identification (noiseless case) undergoes a phase transition and becomes close to 1, for different values of d. This validates Theorem 1. Fixing $\widetilde{C} = 5.6$, we then see that with the total number of queries growing slowly with d, we have successful identification. For the noisy case, the total number of queries is

[8] In the Gaussian white noise model, which is known to be asymptically equivalent to the regression model as $n \to \infty$.

[9] $C_1 = \max_{i \in \mathcal{S}} \|\partial_i f\|_\infty$ and $\alpha = \min_{i \in \mathcal{S}} \|\partial_i f\|_1$

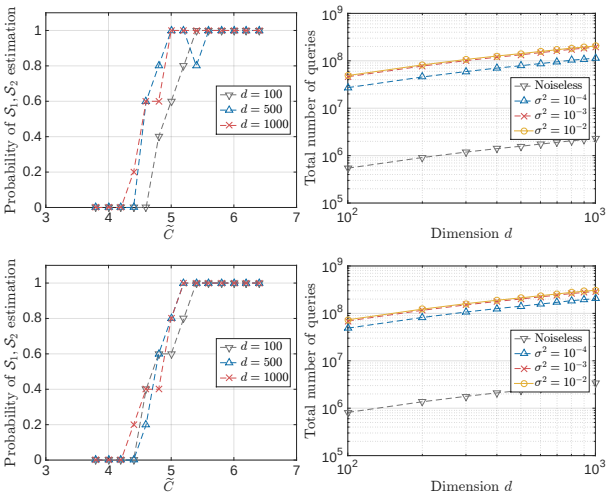


Figure 2: First (resp. second) row is for f_1 (resp. f_2). Left panel depicts the success probability of identifying exactly $\mathcal{S}_1, \mathcal{S}_2$, in the noiseless case. x-axis represent the constant $\widetilde{C}$. The right panel depicts total queries vs. d for exact recovery, with $\widetilde{C} = 5.6$ and various noise settings. All results are over 5 independent Monte Carlo trials.

roughly 10^2 times that in the noiseless setting, however the scaling with d is similar to the noiseless case.

We next consider the relatively harder model: f_2, where the problem parameters are set to: $\lambda_1 = \lambda_2 = 0.3$, $D_1 = 8$, $D_2 = 4$, $B_3 = 35$ and, $m_x = m'_x = 4$. We see in Fig. 2, a phase transition (noiseless case) at $\widetilde{C} = 5.6$ thus validating Theorem 1. For noisy cases, we consider σ^2 as before, and $(N_1, N_2) \in \{(60, 30), (90, 40), (95, 43)\}$. The number of queries is seen to be slightly larger than that for f_1.

Dependence on k. We now demonstrate the scaling of the total number of queries versus the sparsity k for identification of $\mathcal{S}_1, \mathcal{S}_2$. Consider the model $f_3(\mathbf{x}) = \sum_{i=1}^{T}(\alpha_1 \mathbf{x}_{(i-1)5+1} - \alpha_2 \mathbf{x}^2_{(i-1)5+2} + \alpha_3 \mathbf{x}_{(i-1)5+3}\mathbf{x}_{(i-1)5+4} - \alpha_4 \mathbf{x}_{(i-1)5+4}\mathbf{x}_{(i-1)5+5})$ where $\mathbf{x} \in \mathbb{R}^d$ for $d = 500$. Here, $\alpha_i \in [2, 5], \forall i$; *i.e.*, we randomly selected α_i's within range and kept the values fixed for all 5 Monte Carlo iterations. Note that $\rho_m = 2$ and the sparsity $k = 5T$; we consider $T \in \{1, 2, \ldots, 10\}$. We set $\lambda_1 = 0.3$, $\lambda_2 = 1$, $D_1 = 2$, $D_2 = 3$, $B_3 = 6$ and $\widetilde{C} = 5.6$. For the noisy cases, we consider σ^2 as before, and choose the same values for (N_1, N_2) as for f_1. In Figure 3 we see that the number of queries scales as $\sim k \log(d/k)$, and is roughly 10^2 more in the noisy case as compared to the noiseless setting.

Dependence on ρ_m. We now demonstrate the scaling of the total queries versus the maximum degree ρ_m for identification of $\mathcal{S}_1, \mathcal{S}_2$. Consider the model $f_4(\mathbf{x}) = \alpha_1 \mathbf{x}_1 - \alpha_2 \mathbf{x}_2^2 + \sum_{i=1}^{T}(\alpha_{3,i}\mathbf{x}_3\mathbf{x}_{i+3}) + \sum_{i=1}^{5}(\alpha_{4,i}\mathbf{x}_{2+2i}\mathbf{x}_{3+2i})$. We choose $d = 500$, $\widetilde{C} = 6$, $\alpha_i \in [2, \ldots, 5], \forall i$ (as earlier) and set $\lambda_1 = 0.3$, $\lambda_2 = 1$, $D_1 = 2$, $D_2 = 3$, $B_3 = 6$. For $T \geq 2$, we have $\rho_m = T$; we choose $T \in \{2, 3, \ldots, 10\}$. Also note that $k = 13$ throughout. For the noisy cases, we consider σ^2 as before, and choose $(N_1, N_2) \in \{(70, 40), (90, 50), (100, 70)\}$. In Figure 4, we see that the number of queries scales as $\sim \rho_m \log(d/\rho_m)$, and is roughly 10^2 more in the noisy case as compared to the noiseless setting.

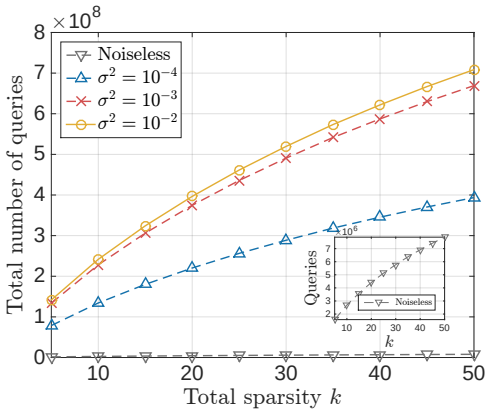


Figure 3: Total number of queries versus k for f_3. This is shown for both noiseless and noisy cases (i.i.d Gaussian).

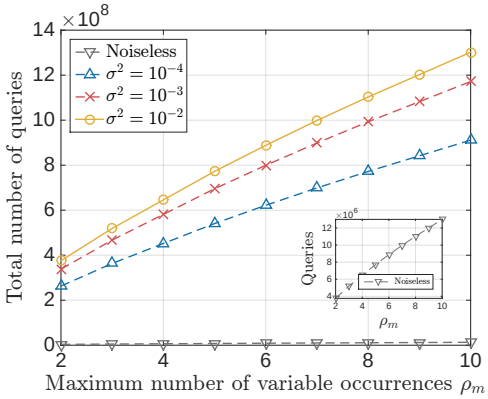


Figure 4: Total number of queries versus ρ_m for f_4. This is shown for both noiseless and noisy cases (i.i.d Gaussian).

8 Concluding remarks

We proposed a sampling scheme for learning a generalized SPAM and provided finite sample bounds for recovering the underlying structure of such models. We also considered the setting where the point queries are corrupted with noise and analyzed sampling conditions for the same. It would be interesting to improve the sampling bounds that we obtained, and under similar assumptions. We leave this for future work.

Acknowledgements. This research was supported in part by SNSF grant CRSII2_147633.

References

[1] Th. Muller-Gronbach and K. Ritter. Minimal errors for strong and weak approximation of stochastic differential equations. *Monte Carlo and Quasi-Monte Carlo Methods*, pages 53–82, 2008.

[2] M.H. Maathuis, M. Kalisch, and P. Bühlmann. Estimating high-dimensional intervention effects from observational data. *Ann. Statist.*, 37(6A):3133–3164, 2009.

[3] M.J. Wainwright. Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. *IEEE Trans. Inform. Theory*, 55(12):5728–5741, 2009.

[4] J.F. Traub, G.W. Wasilkowski, and H. Wozniakowski. *Information-Based Complexity*. Academic Press, New York, 1988.

[5] R. DeVore, G. Petrova, and P. Wojtaszczyk. Approximation of functions of few variables in high dimensions. *Constr. Approx.*, 33:125–143, 2011.

[6] A. Cohen, I. Daubechies, R.A. DeVore, G. Kerkyacharian, and D. Picard. Capturing ridge functions in high dimensions from point queries. *Constr. Approx.*, pages 1–19, 2011.

[7] H. Tyagi and V. Cevher. Active learning of multi-index function models. In *Advances in Neural Information Processing Systems 25*, pages 1466–1474. 2012.

[8] M. Fornasier, K. Schnass, and J. Vybíral. Learning functions of few arbitrary linear parameters in high dimensions. *Foundations of Computational Mathematics*, 12(2):229–262, 2012.

[9] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030, 2009.

[10] L. Meier, S. Van De Geer, and P. Bühlmann. High-dimensional additive modeling. *Ann. Statist.*, 37(6B):3779–3821, 2009.

[11] J. Huang, J.L. Horowitz, and F. Wei. Variable selection in nonparametric additive models. *Ann. Statist.*, 38(4):2282–2313, 2010.

[12] G. Raskutti, M.J. Wainwright, and B. Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *J. Mach. Learn. Res.*, 13(1):389–427, 2012.

[13] H. Tyagi, A. Krause, and B. Gärtner. Efficient sampling for learning sparse additive models in high dimensions. In *Advances in Neural Information Processing Systems 27*, pages 514–522. 2014.

[14] Y. Lin and H.H. Zhang. Component selection and smoothing in multivariate nonparametric regression. *Ann. Statist.*, 34(5):2272–2297, 2006.

[15] P. Radchenko and G. M. James. Variable selection using adaptive nonlinear interaction structures in high dimensions. *J. Amer. Statist. Assoc.*, 105:1541–1553, 2010.

[16] C. B. Storlie, H. D. Bondell, B. J. Reich, and H. H. Zhang. Surface estimation, variable selection, and the nonparametric oracle property. *Statistica Sinica*, 21(2):679–705, 2011.

[17] C. Gu. *Smoothing Spline ANOVA Models*. Springer (New York), 2002.

[18] E.J. Candès, J.K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[19] D.L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, 2006.

[20] P. Wojtaszczyk. ℓ_1 minimization with noisy data. *SIAM J. Numer. Anal.*, 50(2):458–467, 2012.

[21] E. Mossel, R. O'Donnell, and R.P. Servedio. Learning juntas. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, pages 206–212, 2003.

[22] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[23] A.S. Bandeira, K. Scheinberg, and L.N. Vicente. Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization. *Mathematical Programming*, 134(1):223–257, 2012.

[24] C. de Boor. *A practical guide to splines*. Springer Verlag (New York), 1978.

[25] C.J. Stone. The use of polynomial splines and their tensor products in multivariate function estimation. *Ann. Statist.*, 22(1):118–171, 1994.

[26] M. Yuan. Nonnegative garrote component selection in functional anova models. In *AISTATS*, volume 2, pages 660–666, 2007.

[27] V. Koltchinskii and M. Yuan. Sparse recovery in large ensembles of kernel machines. In *COLT*, pages 229–238, 2008.

[28] V. Koltchinskii and M. Yuan. Sparsity in multiple kernel learning. *Ann. Statist.*, 38(6):3660–3695, 2010.

[29] A. Dalalyan, Y. Ingster, and A.B. Tsybakov. Statistical inference in compound functional models. *Probability Theory and Related Fields*, 158(3-4):513–532, 2014.

[30] N.H. Choi, W. Li, and J. Zhu. Variable selection with the strong heredity constraint and its oracle property. *J. Amer. Statist. Assoc.*, 105(489):354–364, 2010.

[31] J. Bien, J. Taylor, and R. Tibshirani. A lasso for hierarchical interactions. *Ann. Statist.*, 41(3):1111–1141, 2013.

[32] B. Nazer and R.D. Nowak. Sparse interactions: Identifying high-dimensional multilinear systems via compressed sensing. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 1589–1596, 2010.

[33] K. Schnass and J. Vybiral. Compressed learning of high-dimensional sparse functions. In *ICASSP*, 2011.

[34] L. Comminges and A.S. Dalalyan. Tight conditions for consistent variable selection in high dimensional nonparametric regression. *J. Mach. Learn. Res.*, 19:187–206, 2012.

[35] L. Comminges and A.S. Dalalyan. Tight conditions for consistency of variable selection in the context of high dimensionality. *Ann. Statist.*, 40(5):2667–2696, 2012.

[36] A. Kyrillidis and V. Cevher. Recipes on hard thresholding methods. In *CAMSAP*, 2011.