Nearly Tight Bounds on ℓ_1 Approximation of Self-Bounding Functions

Vitaly Feldman
IBM Research - Almaden

Pravesh Kothari
UT Austin

Jan Vondrák
IBM Research - Almaden

Abstract

We study the complexity of learning and approximation of self-bounding functions over the uniform distribution on the Boolean hypercube $\{0,1\}^n$. Informally, a function $f : \{0,1\}^n \to \mathbb{R}$ is self-bounding if for every $x \in \{0,1\}^n$, $f(x)$ upper bounds the sum of all the n marginal decreases in the value of the function at x. Self-bounding functions include such well-known classes of functions as submodular and fractionally-subadditive (XOS) functions. They were introduced by Boucheron *et al.* in the context of concentration of measure inequalities [BLM00]. Our main result is a nearly tight ℓ_1-approximation of self-bounding functions by low-degree juntas. Specifically, all self-bounding functions can be ϵ-approximated in ℓ_1 by a polynomial of degree $\tilde{O}(1/\epsilon)$ over $2^{\tilde{O}(1/\epsilon)}$ variables. Both the degree and junta-size are optimal up to logarithmic terms. Previously, the best known bound was $O(1/\epsilon^2)$ on the degree and $2^{O(1/\epsilon^2)}$ on the number of variables [FV13]. These results lead to improved and in several cases almost tight bounds for PAC and agnostic learning of submodular, XOS and self-bounding functions. In particular, assuming hardness of learning juntas, we show that PAC and agnostic learning of self-bounding functions have complexity of $n^{\tilde{\Theta}(1/\epsilon)}$.

1 Introduction

We consider learning and approximation of several classes of real-valued functions over the uniform distribution on the Boolean hypercube $\{0,1\}^n$. The most well-studied class of functions that we consider is the class of submodular functions. Submodularity, a discrete analog of convexity, has played an essential role in combinatorial optimization [Lov83]. It appears in many important settings including cuts in graphs [GW95, Que95, FFI01], rank function of matroids [Edm, Fra97], set covering problems [Fei98], and plant location problems [CFN77]. A related class of functions is that of *fractional subadditive functions*, equivalently known as XOS functions, which generalize monotone submodular functions and have been introduced in the context of combinatorial auctions [BLN06]. It turns out that these classes are all contained in a broader class, that of *self-bounding functions*, introduced in the context of concentration of measure inequalities [BLM00]. Informally, a function f over $\{0,1\}^n$ is a-self-bounding if for every $x \in \{0,1\}^x$, $a \cdot f(x)$ upper bounds the sum of all the n marginal decreases in the value of the function at x. For XOS functions $a = 1$ and for submodular[1] $a = 2$ (a is omitted when it equals 1). See Sec. 2 for formal definitions and examples of self-bounding functions.

Wide-spread applications of submodular functions have recently inspired the question of whether and how such functions can be learned from random examples (of an unknown submodular function). The question was first formally considered by Balcan and Harvey [BH12] who motivate it by learning of valuations functions. Reconstruction of such functions up to some multiplicative factor from value queries (which allow the learner to ask for the value of the function at any point) was also considered by Goemans

[1]Technically, self-bounding functions are always non-negative and hence capture only non-negative submodular functions. Submodularity is preserved under shifting of the function and therefore it is sufficient to consider non-negative submodular functions.

et al. [GHIM09]. In this work we consider the setting in which the learner gets random and uniform examples of an unknown function f and its goal is to find a hypothesis function h that ϵ-approximates the unknown function for a given $\epsilon > 0$. The measure of the approximation error we use is the standard absolute error or ℓ_1-distance, which equals $\mathbf{E}_{x\sim D}[|f(x) - h(x)|]$. While other measures of error, such as ℓ_2, are often studied in machine learning, there is a large number of scenarios where the expected absolute error is used. For example, if the unknown function is Boolean then learning with ℓ_1 error is equivalent to learning with Boolean disagreement error [KKMS08]. Applications of learning algorithms for submodular functions to differentially-private data release require ℓ_1 error [GHRU11, CKKL12, FK13] as does learning of probabilistic concepts (which are concepts expressing the probability of an event) [KS94].

Motivated by applications to learning, prior works have also studied a number of natural questions on approximation of submodular and related classes of functions by concisely represented functions. For example, linear functions [BH12], low-degree polynomials [CKKL12], DNF formulas [RY13], decision trees [FKV13] and functions of few variables (referred to as *juntas*) [FKV13, BOSY13, FV13]. We survey the prior work in more detail in Section 1.2.

1.1 Our Results

In this work, we provide nearly tight bounds on approximation of self-bounding functions by low-degree polynomials and juntas in the ℓ_1-norm. Previous approximation bounds for the uniform distribution relied on bounding ℓ_2 error that is more convenient to analyze using Fourier techniques. However this approach has so far led to weaker bounds on ℓ_1 approximation error. The dependence of the degree and junta size on the error parameter ϵ in our bounds is quadratically better (up to a logarithmic term) than bounds which are known for ℓ_2 error.

Structural results: Our two key structural results can be summarized as follows.

Theorem 1.1 *Let $f : \{0,1\}^n \to [0,1]$ be an a-self-bounding function and $\epsilon > 0$. For $d = O(a/\epsilon \cdot \log(1/\epsilon))$ there exists a set of indices I of size $2^{O(d)}$ and a polynomial p of degree d over variables in I such that $\|f - p\|_1 \leq \epsilon$.*

This result itself is based on a combination of two structural results. The first one gives a degree bound of $O(\frac{a}{\epsilon}\log\frac{a}{\epsilon})$. Previously, it was known that self-bounding functions with range $[0,1]$ can be ϵ-approximated by polynomials of degree $O(1/\epsilon^2)$ [CKKL12]. Alternative proofs of this result were also given in [FKV13] and [FV13]. Our proof is based on the analysis of the function obtained by applying a noise operator to an a-self-bounding function f. This analysis is a generalization of the analysis from [CKKL12] for submodular functions. However, instead of using the standard connection between noise stability and ℓ_2 approximation by polynomials (as done in [CKKL12]) we argue directly about the ℓ_1 error of the polynomial obtained from the noisy version of f. Our analysis also implies that for every non-negative a-self-bounding function f, $\|f\|_1 \geq \frac{1}{3^a}\|f\|_\infty$ (see Lemma 3.4). This has been known for submodular [FMV07] and XOS [Fei06] functions (with a constant a) and, together with approximation by a junta, can be used to obtain a learning algorithm with multiplicative approximation guarantees for all a-self-bounding functions [FV13].

The second component of this result builds on the work of [FV13], where it was shown that a classic theorem of Friedgut [Fri98], on approximation of Boolean functions by juntas, generalizes to the setting of real-valued functions by including a dependence on ℓ_1 as well as ℓ_2-influences of the function. We show that by applying the analysis from [FV13] to the noisy version of f (for which we have better degree bounds) we can obtain approximation by a junta of size $2^{O(a/\epsilon\cdot\log(1/\epsilon))}$. This improves on $2^{O(a/\epsilon^2)}$ bound in [FV13] (that holds also for ℓ_2 error).

Algorithmic applications: It is easy to exploit our structural results in existing algorithms to obtain better running time and sample complexity bounds. We describe two of these results here and some

additional ones in Section 4. The first one is the application to PAC learning of submodular functions from random and uniform examples.

Theorem 1.2 *Let $\mathcal{C}_s$ be the set of all submodular functions from $\{0,1\}^n$ to $[0,1]$. There exists an algorithm $\mathcal{A}$ that given $\epsilon > 0$ and access to random uniform examples of any $f \in \mathcal{C}_s$, with probability at least $2/3$, outputs a function h, such that $\|f - h\|_1 \leq \epsilon$. Further, $\mathcal{A}$ runs in time $\tilde{O}(n^2) \cdot 2^{\tilde{O}(1/\epsilon)}$ and uses $2^{\tilde{O}(1/\epsilon)} \log n$ examples.*

This result follows from using our degree bound with the junta size bound and the influential-variable identification algorithm from [FKV13]. These running time and sample complexity bounds improve on a sequence of previous bounds for the problem [GHRU11, CKKL12, FKV13, FV13] with the strongest previous bound being $\tilde{O}(n^2) \cdot 2^{\tilde{O}(1/\epsilon^2)}$ time and $2^{\tilde{O}(1/\epsilon^2)} \log n$ examples [FV13]. An information theoretic lower bound of $2^{\tilde{O}(1/\epsilon^{2/3})}$ examples is also known for the problem [FKV13].

A second application is the algorithm for learning all a-self-bounding functions in the substantially more challenging, agnostic framework. An agnostic learning algorithm for a class of functions $\mathcal{C}$ is an algorithm that given random examples of *any* function f finds a hypothesis h whose error is at most ϵ-greater than the error of the best hypothesis in $\mathcal{C}$ (see [KSS94] for the Boolean case).

Theorem 1.3 *Let $\mathcal{C}_a$ be the class of all a-self-bounding functions from $\{0,1\}^n$ to $[0,1]$. There exists an algorithm $\mathcal{A}$ that given $\epsilon > 0$ and access to random uniform examples of any real-valued f, with probability at least $2/3$, outputs a function h, such that $\|f - h\|_1 \leq \Delta + \epsilon$, where $\Delta = \min_{g \in \mathcal{C}_a}\{\|f - g\|_1\}$. Further, $\mathcal{A}$ runs in time $n^{\tilde{O}(a/\epsilon)}$ and uses $2^{\tilde{O}(a^2/\epsilon^2)} \log n$ examples.*

This algorithm is based on polynomial ℓ_1 regression with an additional constraint on the spectral norm of the solution to obtain a stronger sample complexity bound [FV13]. The best previous bound of $n^{O(a/\epsilon^2)}$ time and $2^{O(a^2/\epsilon^4)} \log n$ examples follows from the results in [FV13] for function of low total influence. For submodular functions the sample complexity can be further strengthened to $2^{\tilde{O}(1/\epsilon)} \log n$ examples (see Cor. 3.8). Further details of algorithmic applications are given in Section 4.

Lower bounds: We prove that a-self-bounding functions require degree $\Omega(a/\epsilon)$ to ϵ-approximate in ℓ_1 distance (see Cor. 5.6). A construction of a parity function correlated with a submodular function in [FKV13] also implies that even submodular functions require polynomials of degree $\Omega(\epsilon^{-2/3})$ to ϵ-approximate in ℓ_1.

In [FV13] it is shown that XOS functions require a junta of size $2^{\Omega(1/\epsilon)}$ to ϵ-approximate (however submodular functions admit approximation by exponentially smaller juntas [FV13]). This also implies $2^{\Omega(a/\epsilon)}$ lower bound on junta size for a-self-bounding functions (see Lem. 5.2). Therefore our structural results are essentially tight for self-bounding functions.

We then show that our agnostic learning algorithm for a-self-bounding function is nearly optimal. In fact, even PAC learning of non-monotone a-self-bounding functions requires time $n^{\Omega(a/\epsilon)}$ assuming hardness of learning k-term DNF to accuracy $1/4$ in time $n^{\Omega(k)}$. This is in contrast to the submodular (Thm. 1.2) and monotone self-bounding cases (Thm. 4.4).

Theorem 1.4 *For every $a \geq 1$, if there exists an algorithm that PAC learns a-self-bounding functions with range $[0,1]$ to ℓ_1 error of $\epsilon > 0$ in time $T(n, 1/\epsilon)$ then there exists an algorithm that PAC learns k-DNF formulas to accuracy ϵ' in time $T(n, k/(a \cdot \epsilon'))$ for some fixed constant c.*

To prove this hardness results we show that a k-DNF formula (of any size) is a k-self-bounding function. Using an additional "lifting" trick we can also embed k-DNF formulas into a-self-bounding functions for any $a \geq 1$. Note that any k-junta can be computed by a k-DNF formula. Learning of DNF expressions is a

well-studied problem in learning theory but there are no algorithms for this problem better than the trivial $O(n^k)$ algorithm, even for a constant $\epsilon' = 1/4$. The (potentially simpler) problem of learning k-juntas is also considered very hard [BL97, Blu03]. Until recently, the only non-trivial algorithm for the problem was the $O(n^{0.7k})$-time algorithm by Mossel *et al.* [MOS04]. The best known upper bound is $O(n^{0.6k})$ and was given in the recent breakthrough result of Valiant [Val12]. Learning of k-juntas is also known to have complexity of $n^{\Omega(k)}$ for all statistical query algorithms [BFJ+94]. Theorem 1.4 implies that PAC learning of a-self-bounding functions in time $n^{o(a/\epsilon)}$ would lead to a $n^{o(k)}$ algorithm for learning k-DNF to any constant accuracy and, in particular, an algorithm for PAC learning k-juntas in time $n^{o(k)}$. We note that the dependence on a/ϵ in our lower bound matches our upper bound up to a logarithmic factor.

Finally, we remark that our reduction to learning of k-DNF also implies that PAC learning of a-self-bounding functions requires at least $2^{\Omega(a/\epsilon)}$ random examples or even stronger value queries (see Cor. 5.5). Therefore sample complexity bounds we give are also close to optimal. Further details of lower bounds are given in Section 5.

1.2 Related Work

Below we briefly mention some of the other related work. We direct the reader to [BH12] and [FKV13] for more detailed surveys. Balcan and Harvey study learning of submodular functions without assumptions on the distribution and also require that the algorithm output a value which is within a multiplicative approximation factor of the true value with probability $\geq 1 - \epsilon$ (the model is referred to as *PMAC learning*). This is a very demanding setting and indeed one of the main results in [BH12] is a factor-$\sqrt[3]{n}$ inapproximability bound for submodular functions. This notion of approximation is also considered in subsequent works of Badanidiyuru *et al.* and Balcan *et al.* [BDF+12, BCIW12] where upper and lower approximation bounds are given for other related classes of functions such as XOS and subadditive. We emphasize that these strong lower bounds rely on a very specific distribution concentrated on a sparse set of points, and show that this setting is very different from uniform/product distributions which are the focus of this paper.

In [GHRU11] learning of submodular functions over the uniform distribution is motivated by problems in differentially-private data release. They show that submodular functions with range $[0, 1]$ are ϵ-approximated by a collection of $n^{O(1/\epsilon^2)}$ ϵ^2-Lipschitz submodular functions. Each ϵ^2-Lipschitz submodular function can be ϵ-approximated by a constant. This leads to a learning algorithm running in time $n^{O(1/\epsilon^2)}$, which however requires value oracle access to the target function, in order to build the collection.

In a recent work, Raskhodnikova and Yaroslavtsev consider learning and testing of submodular functions taking values in the range $\{0, 1, \ldots, k\}$ (referred to as *pseudo-Boolean*) [RY13]. The error of a hypothesis in their framework is the probability that the hypothesis disagrees with the unknown function. They build on the approach from [GHRU11] to show that pseudo-Boolean submodular functions can be expressed as $2k$-DNF and then give a $\mathrm{poly}(n) \cdot k^{O(k \log k/\epsilon)}$-time PAC learning algorithm using value queries. Subsequently, Blais *et al.* proved existence of a junta of size $(k \log(1/\epsilon))^{O(k)}$ and used it to give an algorithm for testing submodularity using $(k \log(1/\epsilon))^{\tilde{O}(k)}$ value queries [BOSY13].

2 Preliminaries

2.1 Submodular, subadditive and self-bounding functions

In this section, we define the relevant classes of functions. We refer the reader to [Von10, FV13] for more details.

Definition 2.1 *A set function $f : 2^N \to \mathbb{R}$ is*

- *monotone, if $f(A) \leq f(B)$ for all $A \subseteq B \subseteq N$.*
- *submodular, if $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$ for all $A, B \subseteq N$.*
- *fractionally subadditive, if $f(A) \leq \sum \beta_i f(B_i)$ whenever $\beta_i \geq 0$ and $\sum_{i:a \in B_i} \beta_i \geq 1$ $\forall a \in A$.*

Submodular functions are not necessarily nonnegative, but in many applications (especially when considering multiplicative approximations), this is a natural assumption. Fractionally subadditive functions are nonnegative by definition (by considering $A = B_1, \beta_1 > 1$). In this paper we work exclusively with functions $f : 2^N \to \mathbb{R}_+$.

Next, we introduce *a-self-bounding functions*. Self-bounding functions were defined by Boucheron, Lugosi and Massart [BLM00] as a unifying class of functions that enjoy strong "dimension-free" concentration properties. Currently this is the most general class of functions known to satisfy such concentration bounds. Self-bounding functions are defined generally on product spaces X^n; here we restrict our attention to the hypercube, so the reader can assume that $X = \{0, 1\}$. We identify functions on $\{0, 1\}^n$ with set functions on $N = [n]$ in a natural way. Here we define a somewhat more general class of a-self-bounding functions, following [MR06].

Definition 2.2 *A function $f : \{0, 1\}^n \to \mathbb{R}$ is a-self-bounding, if for all $x \in \{0, 1\}^n$ and $i \in [n]$,*

$$f(x) - \min_{x_i} f(x) \leq 1$$

and

$$\sum_{i=1}^{n} (f(x) - \min_{x_i} f(x)) \leq a f(x).$$

Useful properties of a-self-bounding functions that are easy verify is that they are closed under taking max operation and closed under taking convex combinations. A particular example of a self-bounding function (related to applications of Talagrand's inequality) is a function with the property of *small certificates*: $f : X^n \to \mathbb{Z}_+$ has small certificates, if it is 1-Lipschitz and whenever $f(x) \geq k$, there is a set of coordinates $S \subseteq [n]$, $|S| = k$, such that if $y|_S = x|_S$, then $f(y) \geq k$. Such functions often arise in combinatorics, by defining $f(x)$ to equal the maximum size of a certain structure appearing in x. In Section 5 we also show that k-DNF formulas are k-self-bounding.

2.2 Fourier Analysis on the Boolean Cube

We rely on the standard Fourier transform representation of real-valued functions over $\{0, 1\}^n$ as linear combinations of parity functions. For $S \subseteq [n]$, the parity function $\chi_S : \{0, 1\}^n \to \{-1, 1\}$ is defined by $\chi_S(x) = (-1)^{\sum_{i \in S} x_i}$. The Fourier expansion of f is given by $f(x) = \sum_{S \subseteq [n]} \hat{f}(S)\chi_S(x)$. The degree of highest degree non-zero Fourier coefficient of f is referred to as the *Fourier degree* of f. Note that Fourier degree of f is exactly the polynomial degree of f when viewed over $\{-1, 1\}^n$ instead of $\{0, 1\}^n$ and therefore it is also equal to the polynomial degree of f over $\{0, 1\}^n$. Let $f : \{0, 1\}^n \to \mathbb{R}$ and $\hat{f} : 2^{[n]} \to \mathbb{R}$ be its Fourier transform.

Definition 2.3 (The noise operator) *For $\rho \in [-1, +1], x \in \{0, 1\}^n$, we define a distribution $N_\rho(x)$ over $y \in \{0, 1\}^n$ by letting $y_i = x_i$ with probability $\frac{1+\rho}{2}$ and $y_i = 1 - x_i$ with probability $\frac{1-\rho}{2}$, independently for each i. The noise operator T_ρ acts on functions $f : \{0, 1\}^n \to \mathbb{R}$, and is defined by*

$$(T_\rho f)(x) = \mathbf{E}_{y \sim N_\rho(x)}[f(y)].$$

The noise stability of f at noise rate ρ is

$$\mathbb{S}_\rho(f) = \langle f, T_\rho f\rangle = \mathbf{E}_{x\sim\Pi}[f(x)T_\rho(x)].$$

In terms of Fourier coefficients, the noise operator acts as $\widehat{T_\rho f}(S) = \rho^{|S|}\hat{f}(S)$. Therefore, noise stability can be written as $\mathbb{S}_\rho(f) = \sum_{S\subseteq[n]} \rho^{|S|}\hat{f}^2(S)$.

Definition 2.4 (Discrete derivatives) *For $x \in \{0,1\}^n$, $b \in \{0,1\}$ and $i \in n$ let $x_{i\leftarrow b}$ denote the vector in $\{0,1\}^n$ that equals to x with i-th coordinate set to b. For a real-valued $f : \{0,1\}^n \to \mathbb{R}$ and indices $i, j \in [n]$ we define, $\partial_i f(x) = \frac{1}{2}(f(x_{i\leftarrow 1}) - f(x_{i\leftarrow -1}))$. We also define $\partial_{i,j} f(x) = \partial_i \partial_j f(x)$.*

Observe that $\partial_i f(x) = \sum_{S\ni i} \hat{f}(S)\chi_{S\setminus\{i\}}(x)$, and $\partial_{i,j} f(x) = \sum_{S\ni i,j} \hat{f}(S)\chi_{S\setminus\{i,j\}}(x)$.

We use several notions of *influence* of a variable on a real-valued function which are based on the standard notion of influence for Boolean functions (e.g. [BOL85, KKL88]).

Definition 2.5 (Influences) *For a real-valued $f : \{0,1\}^n \to \mathbb{R}$, $i \in [n]$, and $\kappa \geq 0$ we define the ℓ_κ^κ-*influence *of variable i as $\mathsf{Inf}_i^\kappa(f) = \|\frac{1}{2}\partial_i f\|_\kappa^\kappa = \mathbf{E}[|\frac{1}{2}\partial_i f|^\kappa]$. We define $\mathsf{Inf}^\kappa(f) = \sum_{i\in[n]} \mathsf{Inf}_i^\kappa(f)$ and refer to it as the* total ℓ_κ^κ-influence *of f.*

3 Structural results

3.1 Approximation of Self-bounding Functions by Low-degree Polynomials

In this section, we study the Fourier transform of a self-bounding function and its approximability by a low-degree polynomial. Our main technical result is the following bound, strengthening and generalizing a similar one proved for submodular functions in [CKKL12].

Theorem 3.1 *For every a-self-bounding function $f : \{0,1\}^n \to \mathbb{R}_+$, $n \geq 4a$, and every $\epsilon > 0$, there exists a multilinear polynomial p of degree $d = \lceil \frac{2a}{\epsilon} \log \frac{3}{\epsilon} \rceil$ such that*

$$\|f - p\|_1 \leq \epsilon \|f\|_2.$$

In particular, the polynomial can be chosen as $p(x) = \sum_{|S|<d} \rho^{|S|}\hat{f}(S)\chi_S(x)$, for $\rho = 1 - \frac{\epsilon}{2a}$.

First, we prove the following lemma on how the noise operator affects a self-bounding function (a generalization of a similar lemma about submodular functions in [CKKL12]).

Lemma 3.2 *For any a-self-bounding function $f : \{0,1\}^n \to \mathbb{R}$ under the uniform distribution, and any $\rho \in [-1,+1], x \in \{0,1\}^n$,*

$$T_\rho f(x) \geq \left(1 - \frac{1-\rho}{2(1-\frac{a-1}{n})}\right)^a f(x).$$

Proof: First, let us observe that the statement of the lemma is invariant under flipping the hypercube $\{0,1\}^n$ along any coordinate: the notion of a-self-bounding functions does not change, the action of the noise operator does not change, and the conclusion of the lemma does not change either. So we can assume without loss of generality that $x = (1,1,\ldots,1)$. We also identify points in $\{0,1\}^n$ with sets $S \subseteq [n]$ by considering $S = \{i : x_i = 1\}$.

Let us average the values of f over levels of sets of constant $|S|$, and define

$$\phi(t) = \mathbf{E}_{|S|=t}[f(S)] = \frac{1}{\binom{n}{t}} \sum_{S:|S|=t} f(S).$$

In particular, $\phi(0) = f(\emptyset) = f(x)$. We claim the following: for every $t = 0, 1, \ldots, n$,

$$\phi(t) \geq \left(1 - \frac{t}{n-a+1}\right)^a \phi(0). \tag{1}$$

Intuitively, if $f(x)$ is a point of high value, the value cannot drop off too quickly as we move away from x. If we prove (1), then we are done: $T_\rho f(x)$ can be viewed as an expectation of $f(S)$ over a distribution where the sets on each level appear with the same probability. The expected cardinality of a set sampled from this distribution is $\mathbf{E}[|S|] = \frac{1-\rho}{2}n$. By convexity of the bound (1), we obtain

$$T_\rho f(x) \geq \phi\left(\frac{1-\rho}{2}n\right) \geq \left(1 - \frac{\frac{1-\rho}{2}n}{n-a+1}\right)^a f(x) = \left(1 - \frac{1-\rho}{2(1-\frac{a-1}{n})}\right)^a f(x).$$

So it remains to prove (1).

We proceed by induction. For $t = 0$, the claim is trivial. Let us assume it holds for t, and consider a set S, $|S| = t$. By the property of a-self-bounding, we have

$$af(S) \geq \sum_{i=1}^{n} (f(S) - \min\{f(S+i), f(S-i)\}) \geq \sum_{i \in [n] \setminus S} (f(S) - f(S+i)).$$

Note that $|[n] \setminus S| = n - t$. By rearranging this inequality, we get

$$(n-t-a)f(S) \leq \sum_{i \in [n] \setminus S} f(S+i).$$

Now let us add up this inequality over all S of size $|S| = t$:

$$(n-t-a) \sum_{|S|=t} f(S) \leq \sum_{|S|=t, i \notin S} f(S+i) = (t+1) \sum_{|S'|=t+1} f(S')$$

because every set S' of size $t+1$ appears $t+1$ times in the penultimate summation. Expressing this inequality in terms of $\phi(t)$, we get

$$(n-t-a)\binom{n}{t}\phi(t) \leq (t+1)\binom{n}{t+1}\phi(t+1),$$

or equivalently

$$\phi(t) \leq \frac{n-t}{n-t-a}\phi(t+1).$$

We replace this by a slightly weaker bound: $\phi(t) \leq (\frac{n-t-a+1}{n-t-a})^a \phi(t+1)$. To see why this holds, consider $(\frac{n-t-a+1}{n-t-a})^a = (1 + \frac{1}{n-t-a})^a \geq 1 + \frac{a}{n-t-a} = \frac{n-t}{n-t-a}$.

By the inductive hypothesis (1), we assume $\phi(t) \geq (\frac{n-a+1-t}{n-a+1})^a \phi(0)$. So we obtain

$$\left(\frac{n-a+1-t}{n-a+1}\right)^a \phi(0) \leq \left(\frac{n-t-a+1}{n-t-a}\right)^a \phi(t+1).$$

This implies the claim (1) for $t+1$:

$$\phi(t+1) \geq \left(\frac{n-a-t}{n-a+1}\right)^a \phi(0) = \left(1 - \frac{t+1}{n-a+1}\right)^a \phi(0).$$

□

Corollary 3.3 *For any a-self-bounding function $f : \{0,1\}^n \to \mathbb{R}$ under the uniform distribution, the noise stability with noise parameter ρ is*

$$\mathbb{S}_\rho(f) \geq \left(1 - \frac{1-\rho}{2(1-\frac{a-1}{n})}\right)^a \|f\|_2^2.$$

In particular, for $a = 1$ (self-bounding functions), we obtain $\mathbb{S}_\rho(f) \geq \frac{1+\rho}{2}\|f|_2^2$. In [CKKL12], an analogous bound on noise stability is used to derive agnostic statistical learning within ℓ_1-error ϵ in time $n^{O(1/\epsilon^2)}$. This result is (implicitly) based on an ℓ_1-approximation of a submodular function by a low-degree polynomial. Here, we prove by a more direct argument that every a-self-bounding function f is well approximated in ℓ_1 by a low-degree polynomial, with an improved dependence on ϵ (which implies agnostic learning in time $n^{O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})}$).

Now we are ready to prove Theorem 3.1.

Proof: Let $\rho \in [-1,+1]$ be a noise parameter to be determined later. Denote $\tau = (1 - \frac{1-\rho}{2(1-(a-1)/n)})^a$. By Lemma 3.2, we have $T_\rho f(x) \geq \tau f(x)$. Therefore, using the triangle inequality,

$$\|T_\rho f - f\|_1 \leq \|T_\rho f - \tau f\|_1 + \|\tau f - f\|_1 = \mathbf{E}[T_\rho f(x) - \tau f(x)] + \mathbf{E}[f(x) - \tau f(x)] = 2(1-\tau)\,\|f\|_1 \qquad (2)$$

using the fact that $T_\rho f - \tau f$ and $f - \tau f$, as well as f itself, are nonnegative functions. On the other hand, since $T_\rho f(x) = \sum_{S\subseteq[n]} \rho^{|S|}\hat{f}(S)\chi_S(x)$, we can estimate the tail of the Fourier expansion as follows: For any d, define $f_{<d}(x) = \sum_{S:|S|<d} \hat{f}(S)\chi_S(x)$, a polynomial of degree at most d. Using Lemma 3.4 twice, we get

$$\|T_\rho f_{<d} - T_\rho f\|_1 = \left\| \sum_{S:|S|\geq d} \rho^{|S|}\hat{f}(S)\chi_S \right\|_1 \leq \left\| \sum_{S:|S|\geq d} \rho^{|S|}\hat{f}(S)\chi_S \right\|_2 \leq \rho^d \|f\|_2. \qquad (3)$$

Combining (2) and (3) by the triangle inequality, we obtain

$$\|T_\rho f_{<d} - f\|_1 \leq \|T_\rho f_{<d} - T_\rho f\|_1 + \|T_\rho f - f\|_1 \leq \rho^d \|f\|_2 + 2(1-\tau)\|f\|_2. \qquad (4)$$

We also used the fact that $\|f\|_1 \leq \|f\|_2$. The function $T_\rho f_{<d}$ is our promised polynomial p. Now, let us set $d = \lceil \frac{2a}{\epsilon}\log\frac{3}{\epsilon}\rceil$ and $\rho = 1 - \frac{\epsilon}{2a}$. Considering $n \geq 4a$, we get $\tau \geq (1-\frac{\epsilon}{3a})^a \geq 1 - \frac{\epsilon}{3}$. By (4), the distance of our polynomial from f is

$$\|T_\rho f_{<d} - f\|_1 \leq \left(1 - \frac{\epsilon}{2a}\right)^d \|f\|_2 + 2(1-\tau)\|f\|_2 \leq \frac{1}{3}\epsilon\|f\|_2 + \frac{2}{3}\epsilon\|f\|_2 \leq \epsilon\|f\|_2.$$

□

Comparison of norms for self-bounding functions. Let us mention here that our bound on the noise operator also implies a bound on the ℓ_1 norm of a self-bounding function, relative to its ℓ_∞ norm.

Lemma 3.4 *For any a-self-bounding function $f : \{0,1\}^n \to \mathbb{R}_+$ under the uniform distribution, with $n \geq 4a$,*

$$\|f\|_1 \leq \|f\|_\infty \leq 3^a \|f\|_1.$$

Proof: Let $\|f\|_\infty = f(x^*)$. Since f is nonnegative and $n \geq 4a$, we have by Lemma 3.2

$$\|f\|_1 = \mathbf{E}[f(x)] = T_0 f(x^*) \geq \left(1 - \frac{1}{2(1-\frac{a-1}{n})}\right)^a f(x^*) \geq \left(1 - \frac{1}{2 \cdot 3/4}\right)^a f(x^*) = \frac{1}{3^a} f(x^*).$$

□

We remark that a factor exponential in a is necessary here. Consider the conjunction function on a variables, $f(x) = x_1 x_2 \cdots x_a$. This is an a-self-bounding function with values in $\{0,1\}$. We have

$$\|f\|_p = (\Pr[f(x) = 1])^{1/p} = 2^{-a/p}.$$

In particular, $\|f\|_1 = 2^{-a}$, $\|f\|_2 = 2^{-a/2}$ and $\|f\|_\infty = 1$; i.e., the ℓ_1, ℓ_2 and ℓ_∞ norms can differ by factors exponential in a.

Relative error vs. additive error. In our results, we typically assume that the values of $f(x)$ are in a bounded interval $[0,1]$ or that $\|f\|_1 \leq 1$ and our goal is to approximate f with an additive error of ϵ. As Lemma 3.4 shows, for a-self-bounding functions (with constant a) the ℓ_1 and ℓ_∞ norms are within a bounded factor, so this does not make much difference.

This means that if we scale $f(x)$ by $1/(3^a\|f\|_1)$, we obtain a function with values in $[0,1]$. Approximating this function within an additive error of ϵ is equivalent to approximating the original function within an error of $\epsilon 3^a \|f\|_1$. In particular, for submodular functions we have $a = 2$. Hence, the two settings are equivalent up to a constant factor in the error and we state our results for submodular functions in the interval $[0,1]$.

3.2 Friedgut's Theorem for ℓ_1-approximation

As we have shown in Lemma 3.6, self-bounding functions have low average sensitivity. A celebrated result of Friedgut [Fri98] shows that any Boolean function on $\{0,1\}^n$ of low average sensitivity is close to a function that depends on few variables. His result was extended to ℓ_2 approximation of real-valued functions in [FV13]. We now show that for self-bounding functions a tighter bounds can be achieved for ℓ_1 approximation. Our proof is based on the use of ℓ_1 approximation by polynomials proved in Theorem 3.1 together with the analysis from [FV13] to obtain a smaller ℓ_1 approximating junta.

We now state Thm. 1.1 in more detail.

Theorem 3.5 (Thm. 1.1 restated) *Let $f : \{0,1\}^n \to [0,1]$ be an a-self-bounding function. For every $\epsilon > 0$, let $d = \lceil \frac{4a}{\epsilon} \log \frac{6}{\epsilon} \rceil$ and $I = \{i \in [n] \mid \mathsf{Inf}_i^{4/3}(f) \geq \alpha\}$ for $\alpha = 3^{-2d-1}\epsilon^4/a^2$. Then $|I| \leq a/\alpha$ and there exists a polynomial p of degree d over variables in I such that $\|f - p\|_1 \leq \epsilon$.*

Our proof will requires two lemmas from [FV13]. The first one shows that self-bounding functions have low total ℓ_1-influence.

Lemma 3.6 *Let $f : \{0,1\}^n \to \mathbb{R}_+$ be an a-self-bounding function. Then $\mathsf{Inf}^1(f) \leq a \cdot \|f\|_1$. In particular, for $f : \{0,1\}^n \to [0,1]$, $\mathsf{Inf}^1(f) \leq a$. For a submodular $f : \{0,1\}^n \to [0,1]$, $\mathsf{Inf}^1(f) \leq 2$; for an XOS $f : \{0,1\}^n \to [0,1]$, $\mathsf{Inf}^1(f) \leq 1$.*

The second lemma is the following bound on the sum of squares of all low-degree Fourier coefficients that include a variable of low influence.

Lemma 3.7 *Let $f : \{0,1\}^n \to \mathbb{R}$, $\kappa \in (1,2)$, $\alpha > 0$ and d be an integer ≥ 1. Let $I = \{i \in [n] \mid \mathsf{Inf}_i^\kappa(f) \geq \alpha\}$. Then*

$$\sum_{S \not\subseteq I, |S| \leq d} \hat{f}(S)^2 \leq (\kappa - 1)^{1-d} \cdot \alpha^{2/\kappa - 1} \cdot \mathsf{Inf}^\kappa(f) \ .$$

We can now complete the proof of Thm. 3.5.

Proof: Theorem 3.1 proves that for $d \leq \lceil \frac{4a}{\epsilon} \log \frac{6}{\epsilon} \rceil$ and $\rho = 1 - \frac{\epsilon}{2a}$, the function $T_\rho f_{<d}$ satisfies

$$\|f - T_\rho f_{<d}\|_1 \leq \epsilon \|f\|_2 / 2 \leq \epsilon/2. \tag{5}$$

We can also apply Lemma 3.7 with $\kappa = 4/3$ and $\alpha = 3^{-2d-1}\epsilon^4/a^2$ to obtain that

$$\sum_{S \not\subseteq I, |S| \leq d} \hat{f}(S)^2 \leq 3^{d-1} \cdot \alpha^{1/2} \cdot \mathsf{Inf}^{4/3}(f) = 3^{d-1} \cdot \left(3^{-d-1/2} \cdot \frac{\epsilon^2}{a}\right) \cdot \mathsf{Inf}^{4/3}(f) \leq \frac{\epsilon^2}{4} \ , \tag{6}$$

where the last inequality uses $\mathsf{Inf}^{4/3}(f) \leq \mathsf{Inf}^1(f) \leq a$ which follows from Lemma 3.6 and the fact that $\partial_i f$'s have range $[-1/2, 1/2]$ when f has range $[0,1]$.

For every S, $|\widehat{T_\rho f}(S)| = |\rho^{|S|} \hat{f}(S)| \leq |\hat{f}(S)|$. Therefore eq. (6) implies that

$$\sum_{S \not\subseteq I, |S| \leq d} \widehat{T_\rho f}(S)^2 \leq \sum_{S \not\subseteq I, |S| \leq d} \hat{f}(S)^2 \leq \frac{\epsilon^2}{4} \ . \tag{7}$$

Now let $p = \sum_{S \subseteq I,\ |S| \leq d} \widehat{T_\rho f}(S) \chi_S$ be the restriction of $T_\rho f_{<d}$ to variables in I. Equation (7) gives a bound on the sum of squares of all the coefficients that we removed from $T_\rho f_{<d}$ and implies that $\|p - T_\rho f_{<d}\|_1 \leq \|p - T_\rho f_{<d}\|_2 \leq \epsilon/2$. Together with eq. (5), we get $\|f - p\|_1 \leq \epsilon$. Finally, $|I| \leq \mathsf{Inf}^{4/3}(f)/\alpha \leq \mathsf{Inf}^1(f)/\alpha \leq a/\alpha$. □

An immediate corollary of Thm. 3.5 is that for every a-self-bounding function there exists a polynomial of low total ℓ_1-spectral norm that approximates it. For a submodular function f we can also use the upper bound of $O(1/\epsilon^2 \cdot \log(1/\epsilon))$ on ℓ_1 approximation by a junta [FV13] to strengthen the upper bound on the spectral norm.

Corollary 3.8 *Let $f : \{0,1\}^n \to [0,1]$ be an a-self-bounding function and $\epsilon > 0$. There exist $d = O(a/\epsilon \cdot \log(1/\epsilon))$ and a polynomial p of degree d such that $\|f - p\|_1 \leq \epsilon$ and $\|\hat{p}\|_1 = 2^{O(d^2)}$, where $\|\hat{p}\|_1 = \sum_{S \subseteq [n]} |\hat{p}(S)|$. In addition, if f is submodular then $\|\hat{p}\|_1 = 2^{\tilde{O}(1/\epsilon)}$.*

4 Algorithmic Applications

We now outline the applications of our structural results. They are based on using our stronger bounds in existing learning algorithms for submodular, XOS and self-bounding functions.

4.1 Learning Models

Our learning algorithms are in one of two standard models of learning. The first one assumes that the learner has access to random examples of an unknown function from a known set of functions. This model can be seen as a generalization of Valiant's PAC learning model to real-valued functions [Val84].

Definition 4.1 (ℓ_1 PAC learning) *Let $\mathcal{F}$ be a class of real-valued functions on $\{0,1\}^n$ and let $\mathcal{D}$ be a distribution on $\{0,1\}^n$. An algorithm $\mathcal{A}$ PAC learns $\mathcal{F}$ on $\mathcal{D}$, if for every $\epsilon > 0$ and any target function $f \in \mathcal{F}$, given access to random independent samples from $\mathcal{D}$ labeled by f, with probability at least $\frac{2}{3}$, $\mathcal{A}$ returns a hypothesis h such that $\mathbf{E}_{x\sim\mathcal{D}}[|f(x) - h(x)|] \leq \epsilon$.*

While in general Valiant's model does not make assumptions on the distribution $\mathcal{D}$, here we only consider the *distribution-specific* version of the model in which the distribution is fixed and is uniform over $\{0,1\}^n$.

Agnostic learning generalizes the definition of PAC learning to scenarios where one cannot assume that the input labels are consistent with a function from a given class [Hau92, KSS94] (for example as a result of noise in the labels).

Definition 4.2 (ℓ_1 agnostic learning) *Let $\mathcal{F}$ be a class of real-valued functions on $\{0,1\}^n$ and let $\mathcal{D}$ be any fixed distribution on $\{0,1\}^n$. For any distribution $\mathcal{D}'$, let $opt(\mathcal{D}', \mathcal{F})$ be defined as:*

$$opt(\mathcal{D}', \mathcal{F}) = \inf_{f\in\mathcal{F}} \mathbf{E}_{(x,y)\sim\mathcal{D}'}[|y - f(x)|].$$

An algorithm $\mathcal{A}$, is said to agnostically learn $\mathcal{F}$ on $\mathcal{D}$ if for every $\epsilon > 0$ and any distribution $\mathcal{D}'$ on $\{0,1\}^n \times \mathbb{R}^n$ such that the marginal of $\mathcal{D}'$ on $\{0,1\}^n$ is $\mathcal{D}$, given access to random independent examples drawn from $\mathcal{D}'$, with probability at least $\frac{2}{3}$, $\mathcal{A}$ outputs a hypothesis h such that $\mathbf{E}_{(x,y)\sim\mathcal{D}'}[|h(x) - y|] \leq opt(\mathcal{D}') + \epsilon$.

4.2 PAC Learning

Our first application is for PAC learning of submodular functions and is given in Theorem 1.2. The proof of this result is based on approximation by $\tilde{O}(1/\epsilon^2)$-junta and an algorithm for finding the influential variables of a submodular function given in [FV13].

Theorem 4.3 ([FV13]) *Let $f : \{0,1\}^n \to [0,1]$ be a submodular function. There exists an algorithm, that given any $\epsilon > 0$ and access to random and uniform examples of f, with probability at least $5/6$, finds a set of variables I of size $\tilde{O}(1/\epsilon^5)$ such that there exists a submodular J-junta h for $J \subseteq I$ of size $\tilde{O}(1/\epsilon^2)$ satisfying $\|f - h\|_1 \leq \epsilon$. The algorithm runs in time $\tilde{O}(n^2/\epsilon^{10})$ and uses $\tilde{O}(\log(n)/\epsilon^{10})$ examples.*

Given the set I we can use the degree bound from Theorem 3.1 to obtain that there is a polynomial of degree $O(1/\epsilon \cdot \log(1/\epsilon))$ over variables in I that ϵ-approximates f in ℓ_1 norm. This implies that we can use polynomial ℓ_1 regression restricted to variables in I to learn f in time $O(n) \cdot 2^{\tilde{O}(1/\epsilon)}$ and using $2^{\tilde{O}(1/\epsilon)}$ examples. Together with the bounds in Thm.4.3 we obtain the proof of Theorem 1.2.

Our second application is PAC learning of monotone self-bounding functions (the results also apply to *unate* functions which are either monotone or anti-monotone in each variable). Note that this class of functions includes XOS functions.

Theorem 4.4 *Let $\mathcal{C}_a^+$ be the set of all monotone a-self-bounding functions on from $\{0,1\}^n$ to $[0,1]$. There exists an algorithm that PAC learns $\mathcal{C}_a^+$ over the uniform distribution, runs in time $\tilde{O}(n) \cdot 2^{\tilde{O}(a^2/\epsilon^2)}$ and uses $2^{\tilde{O}(a^2/\epsilon^2)} \log n$ examples.*

The proof of this result follows from substituting our bounds in Theorems 1.1 and 3.8 into the simple analysis from [FV13].

4.3 Agnostic Learning

Our main application to agnostic learning is the algorithm for learning self-bounding functions from random examples described in Theorem 1.3. The algorithm used to prove this result is again polynomial ℓ_1 regression over all monomials of degree $\tilde{O}(a/\epsilon)$. In addition, we can rely on the existence of a polynomial of low spectral norm to obtain substantially tighter bounds on sample complexity. Namely, as in [FV13], we use the uniform convergence bounds for linear combinations of functions with ℓ_1 constraint on the sum of coefficients [KST08] (without this result the sample complexity would be $n^{\tilde{O}(a/\epsilon)}$.

Our structural results also have immediate implications for learning with value queries, that is oracle access to the value of the unknown function at any point x. Following the approach from [FKV13], we can use the algorithm of Gopalan *et al.* [GKK08] together with our bounds on the spectral norm of the approximating polynomial in Cor. 3.8. This leads to the following algorithms.

Theorem 4.5 *1. Let $\mathcal{C}_a$ be the class of all a-self-bounding functions from $\{0,1\}^n$ to $[0,1]$. There exists an agnostic learning algorithm that given access to value queries learns $\mathcal{C}_a$ over the uniform distribution in time poly$(n) \cdot 2^{\tilde{O}(a^2/\epsilon^2)}$.*

2. Let $\mathcal{C}_s$ be the class of all submodular functions from $\{0,1\}^n$ to $[0,1]$. There exists an agnostic learning algorithm that given access to value queries learns $\mathcal{C}_s$ over the uniform distribution in time poly$(n) \cdot 2^{\tilde{O}(1/\epsilon)}$.

5 Lower Bounds for Learning Self-Bounding Functions

In this section, we show that learning a-self bounding functions within an error of at most ϵ, is at least as hard as learning the class of all DNFs (of any size) of width at most $\lfloor \frac{a}{4\epsilon} \rfloor$ to an accuracy of $\frac{1}{4}$. Our reduction to learning width k-DNFs (also referred to as k-DNFs) is based on the simple observation that k-DNFs are k-self bounding functions combined with a simple linear transformation that reduces approximation and learning of $(a \cdot r)$-self bounding functions for $r \geq 1$ to that of a-self-bounding functions.

Lemma 5.1 *A function $f : \{0,1\}^n \to \{0,1\}$ computed by a k-DNF formula is a k-self bounding function.*

Proof: Since f is $\{0,1\}$-valued, clearly, $f(x) - \min_{x_i} f(x) \leq 1$ for any $i \in [n]$. If $f(x) = 0$, then, $\sum_{i=1}^{n}(f(x) - \min_{x_i} f(x)) = 0 \leq k \cdot f(x)$. Now suppose $f(x) = 1$. Then, there exists at least one term, say T, of the DNF that is satisfied by the assignment x. Observe that if we flip a literal outside of T, then, the value of f remains unchanged. Thus, if the term indexed by j in $\sum_{i=1}^{n}(f(x) - \min_{x_i} f(x))$ contributes the value 1, then either $x_j \in T$ or $\bar{x}_j \in T$. In particular, at most k terms in the sum contribute 1 and the rest contribute 0. Thus, $\sum_{i=1}^{n}(f(x) - \min_{x_i} f(x)) \leq k = k \cdot f(x)$. □

Remark 5.1 *In light of Lemma 5.1 it is natural to ask whether all Boolean k-self-bounding functions are k-DNF. It is easy to see that for Boolean functions being k-self-bounding can be equivalently stated as having 1-sensitivity of k. The smallest k for which f can be represented by a k-DNF is referred to as* 1*-certificate complexity of f. It has long been observed that for monotone functions* 1*-certificate complexity equals* 1*-sensitivity [Nis89] and therefore all monotone k-self-bounding functions are k-DNF. However this is no longer true for non-monotone functions. A simple example in [Nis89] gives a function with a factor two gap between these two measures. Quadratic gap for every k up to $\theta(n^{1/3})$ is also known [Cha05].*

Next, we observe that for any a-self-bounding function, the function g defined by $g(x) = 1 - \frac{1}{r} + \frac{f(x)}{r}$ is $\frac{a}{r}$-self-bounding whenever $r \geq 1$. This "lifting" transforms an a-self-bounding functions into an $\frac{a}{r}$-self-bounding functions.

Lemma 5.2 *Let $f : \{0,1\}^n \to [0,1]$ be an a-self-bounding function. Then for any $r \geq 1$, $g(x) = 1 - \frac{1}{r} + \frac{f(x)}{r}$ has range $[0,1]$ and is $\frac{a}{r}$-self-bounding.*

Proof: Clearly, the $1 - 1/r + f(x)/r$ transformation maps $[0,1]$ to $[1 - 1/r, 1] \subseteq [0,1]$. Observe that for any x and $i \in [n]$, $g(x) - \min_{x_i} g(x) = \frac{1}{r} \cdot (f(x) - \min_{x_i} f(x))$ and also that $g(x) \geq f(x)$. By the definition of a-self-boundedness we obtain that g is a/r-self bounding. □

Observe that given random examples labeled by f, it is easy to simulate random examples labeled by g. Further, ℓ_1-approximation of f within ϵ can be translated (via the same "lifting") to ϵ/r-approximation of g and vice versa. An immediate corollary of this is that one can use a learning algorithm for a/r-self-bounding functions to learn a-self bounding functions. We use $\mathcal{C}_a^n$ to denote the class of all a-self-bounding functions from $\{0,1\}^n$ to $[0,1]$.

Lemma 5.3 *Let $a \geq 1$ and $a \geq r \geq 1$. Suppose there is an algorithm that PAC (or agnostically) learns $\mathcal{C}_{a/r}^n$ over a distribution D with ℓ_1 error of ϵ in time $T(n, 1/\epsilon)$. Then, there is an algorithm that PAC (or, respectively, agnostically) learns $\mathcal{C}_a^n$ over D with ℓ_1 error of ϵ in time $T(n, 1/(r\epsilon))$.*

The simple structural observations above give us our lower bounds for learning and approximation of a-self-bounding functions. Using Lemmas 5.1 and 5.3, we have the the following lower bound on the time required to PAC learn a-self-bounding functions.

Theorem 5.4 (Th. 1.4 restated) *Suppose there exists an algorithm that PAC learns $\mathcal{C}_a^n$ with ℓ_1 error of $\epsilon > 0$ with respect to the uniform distribution in time $T(n, 1/\epsilon)$. Then, for any $k \geq a$, there exists an algorithm that PAC learns k-DNF formulas with disagreement error of at most ϵ' with respect to the uniform distribution in time $T(n, \frac{k}{a\epsilon'})$. Consequently, there exists an algorithm for learning k-juntas on the uniform distribution to an error of at most $1/4$ in time $T(n, \frac{k}{4a})$ for any $k \geq a$.*

Now, k-juntas contain the set of all Boolean functions on any fixed subset of k variables. A standard information-theoretic lower bound implies that any algorithm that PAC learn k-juntas to an accuracy of $1/4$ on the uniform distribution needs $\Omega(2^k)$ random examples or even value queries. This translates into the following unconditional lower bound for learning a-self-bounding functions.

Corollary 5.5 *Any algorithm that PAC learns $\mathcal{C}_a$ over the uniform distribution needs $\Omega(2^{a/\epsilon})$ random examples or value queries.*

Finally, observe that the $\{0,1\}$-valued parity function on k bits is computed by a k-DNF formula and any polynomial that $1/4$-approximates in ℓ_1 distance on the uniform distribution must have degree at least k. Thus, we have the following degree lower bound for polynomials that ℓ_1 approximate a-self-bounding functions on the uniform distribution on $\{0,1\}^n$.

Corollary 5.6 *Fix an $a \geq 1$ and $\epsilon \in (0, 1/4]$. There exists an a-self-bounding function $f : \{0,1\}^n \to [0,1]$, such that every polynomial p that ϵ-approximates f in ℓ_1 norm with respect to the uniform distribution has degree $d \geq a/(4\epsilon)$.*

Proof: Let $k = \frac{a}{4\epsilon}$ (ignoring rounding issues for simplicity) and f be a $\{0,1\}$-valued parity on some set of k variables. By Lemma 5.1 f is k-self-bounding. Then, as in the proof of Lemma 5.3, for $r = \frac{1}{4\epsilon} \geq 1$, g defined by $g(x) = 1 - \frac{1-f(x)}{r}$ is an a-self-bounding function. Let p be a polynomial of degree d that approximates g within an ℓ_1 error of ϵ with respect to the uniform distribution on $\{0,1\}^n$. Then, as in the proof of Lemma 5.3, $p' = 1 - r(1-p)$ is a polynomial of degree d and approximates f within an ℓ_1 error of at most $\frac{1}{4\epsilon} \cdot \epsilon = 1/4$.

For the $\{-1,1\}$-valued parity $\chi = 2f(x) - 1$ and any polynomial p' of degree less than k, $\mathbf{E}[\chi \cdot p'] = 0$. Further, $\mathbf{E}[|\chi - p'|] \geq 1 - \mathbf{E}[\chi \cdot p'] = 1$. This implies that for f the ℓ_1 error of any polynomial of degree at most $k-1$ is at least $1/2$. In particular, $d \geq a/(4\epsilon)$. □

We remark that slightly weaker versions of Cor. 5.5 and Cor. 5.6 are known for monotone submodular functions. Specifically they require $2^{\Omega(\epsilon^{-2/3})}$ random examples or value queries to PAC learn and also degree $\Omega(\epsilon^{-2/3})$ to approximate [FKV13]. This suggests a natural open problem of whether the lower bounds for submodular functions can be strengthened to $2^{\Omega(1/\epsilon)}$ examples and $\Omega(1/\epsilon)$ degree.

References

[BCIW12] M.F. Balcan, F. Constantin, S. Iwata, and L. Wang. Learning valuation functions. *COLT*, 23:4.1–4.24, 2012.

[BDF+12] A. Badanidiyuru, S. Dobzinski, Hu Fu, R. Kleinberg, N. Nisan, and T. Roughgarden. Sketching valuation functions. In *SODA*, pages 1025–1035, 2012.

[BFJ+94] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of STOC*, pages 253–262, 1994.

[BH12] M.F. Balcan and N. Harvey. Submodular functions: Learnability, structure, and optimization. *CoRR*, abs/1008.2159, 2012. Earlier version in STOC 2011.

[BL97] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[BLM00] S. Boucheron, G. Lugosi, and P. Massart. A sharp concentration inequality with applications. *Random Struct. Algorithms*, 16(3):277–292, 2000.

[BLN06] D. J. Lehmann B. Lehmann and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55:1884–1899, 2006.

[Blu03] A. Blum. Open problem: Learning a function of r relevant variables. In *COLT*, pages 731–733, 2003.

[BOL85] M. Ben-Or and N. Linial. Collective coin flipping, robust voting schemes and minima of banzhaf values. In *FOCS*, pages 408–416, 1985.

[BOSY13] E. Blais, K. Onak, R. Servedio, and G. Yaroslavtsev. Concise representations of discrete submodular functions, 2013. Personal communication.

[CFN77] G. Cornuejols, M. Fisher, and G. Nemhauser. Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. *Management Science*, 23:789–810, 1977.

[Cha05] Sourav Chakraborty. Sensitivity, block sensitivity and certificate complexity of boolean functions, 2005.

[CKKL12] M. Cheraghchi, A. Klivans, P. Kothari, and H. Lee. Submodular functions are noise stable. In *SODA*, pages 1586–1592, 2012.

[Edm] Jack Edmonds. Matroids, submodular functions and certain polyhedra. *Combinatorial Structures and Their Application*.

[Fei98] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[Fei06] Uriel Feige. On maximizing welfare when utility functions are subadditive. In *ACM STOC*, pages 41–50, 2006.

[FFI01] L. Fleischer, S. Fujishige, and S. Iwata. A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions. *JACM*, 48(4):761–777, 2001.

[FK13] V. Feldman and P. Kothari. Learning coverage functions. *arXiv, CoRR*, abs/1304.2079, 2013.

[FKV13] V. Feldman, P. Kothari, and J. Vondrák. Representation, approximation and learning of submodular functions using low-rank decision trees. *COLT*, 2013.

[FMV07] U. Feige, V. Mirrokni, and J. Vondrák. Maximizing non-monotone submodular functions. In *IEEE FOCS*, pages 461–471, 2007.

[Fra97] András Frank. Matroids and submodular functions. *Annotated Biblographies in Combinatorial Optimization*, pages 65–80, 1997.

[Fri98] E. Friedgut. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica*, 18(1):27–35, 1998.

[FV13] Vitaly Feldman and Jan Vondrák. Optimal bounds on approximation of submodular and xos functions by juntas. In *FOCS*, pages 227–236, 2013.

[GHIM09] M. Goemans, N. Harvey, S. Iwata, and V. Mirrokni. Approximating submodular functions everywhere. In *SODA*, pages 535–544, 2009.

[GHRU11] A. Gupta, M. Hardt, A. Roth, and J. Ullman. Privately releasing conjunctions and the statistical query barrier. In *STOC*, pages 803–812, 2011.

[GKK08] P. Gopalan, A. Kalai, and A. Klivans. Agnostically learning decision trees. In *STOC*, pages 527–536, 2008.

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[Hau92] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

[KKL88] J. Kahn, G. Kalai, and N. Linial. The influence of variables on Boolean functions. In *FOCS*, pages 68–80, 1988.

[KKMS08] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

[KS94] M. Kearns and R. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48:464–497, 1994.

[KSS94] M. Kearns, R. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

[KST08] S. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NIPS*, pages 793–800, 2008.

[Lov83] László Lovász. Submodular functions and convexity. *Mathematical Programmming: The State of the Art*, pages 235–257, 1983.

[MOS04] E. Mossel, R. O'Donnell, and R. Servedio. Learning functions of k relevant variables. *JCSS*, 69(3):421–434, 2004.

[MR06] C. McDiarmid and B. Reed. Concentration for self-bounding functions and an inequality of talagrand. *Random structures and algorithms*, 29:549–557, 2006.

[Nis89] N. Nisan. Crew prams and decision trees. In *STOC*, pages 327–335, 1989.

[Que95] Maurice Queyranne. A combinatorial algorithm for minimizing symmetric submodular functions. In *SODA*, pages 98–101, 1995.

[RY13] S. Raskhodnikova and G. Yaroslavtsev. Learning pseudo-boolean k-DNF and submodular functions. In *SODA*, 2013.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Val12] G. Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *The 53rd Annual IEEE Symposium on the Foundations of Computer Science (FOCS)*, 2012.

[Von10] J. Vondrák. A note on concentration of submodular functions, 2010. arXiv:1005.2791v1.