Optimal Bounds on Approximation of Submodular and XOS Functions by Juntas

Vitaly Feldman
IBM Research - Almaden
San Jose, CA, USA
Email: vitaly@post.harvard.edu

Jan Vondrák
IBM Research - Almaden
San Jose, CA, USA
Email: jvondrak@us.ibm.com

***Abstract*—We investigate the approximability of several classes of real-valued functions by functions of a small number of variables (*juntas*). Our main results are tight bounds on the number of variables required to approximate a function $f : \{0,1\}^n \to [0,1]$ within ℓ_2-error ϵ over the uniform distribution:**

- **If f is submodular, then it is ϵ-close to a function of $O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$ variables. This is an exponential improvement over previously known results [1]. We note that $\Omega(\frac{1}{\epsilon^2})$ variables are necessary even for linear functions.**
- **If f is fractionally subadditive (XOS) it is ϵ-close to a function of $2^{O(1/\epsilon^2)}$ variables. This result holds for all functions with low total ℓ_1-influence and is a real-valued analogue of Friedgut's theorem for boolean functions. We show that $2^{\Omega(1/\epsilon)}$ variables are necessary even for XOS functions.**

As applications of these results, we provide learning algorithms over the uniform distribution. For XOS functions, we give a PAC learning algorithm that runs in time $2^{1/\mathrm{poly}(\epsilon)}\mathrm{poly}(n)$. For submodular functions we give an algorithm in the more demanding PMAC learning model [2] which requires a multiplicative $(1+\gamma)$ factor approximation with probability at least $1-\epsilon$ over the target distribution. Our uniform distribution algorithm runs in time $2^{1/\mathrm{poly}(\gamma\epsilon)}\mathrm{poly}(n)$. This is the first algorithm in the PMAC model that can achieve a constant approximation factor arbitrarily close to 1 for all submodular functions (even over the uniform distribution). It relies crucially on our approximation by junta result. As follows from the lower bounds in [1] both of these algorithms are close to optimal. We also give applications for proper learning, testing and agnostic learning with value queries of these classes.

***Index Terms*—submodular; fractionally-subadditive; approximation; junta; PAC learning; testing**

I. Introduction

In this paper, we study the structure and learnability of several classes of real-valued functions over the uniform distribution on the Boolean hypercube $\{0,1\}^n$. The primary 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 [3], [4], [5]. Recently, interest in submodular functions has been revived by new applications in algorithmic game theory as well as machine learning. In machine learning, several applications [6], [7] have relied on the fact that the information provided by a collection of sensors is a submodular function. In algorithmic game theory, submodular functions have found application as *valuation functions* with the property of diminishing returns [8]. Along with submodular functions, other related classes have been studied in the context of algorithmic game theory context: coverage functions, gross substitutes, fractionally subadditive (XOS) functions, etc. 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 [9]. We refer the reader to Section II for definitions and relationships of these classes.

Our focus in this paper is on *structural properties* of these classes of functions, specifically on their approximability by *juntas* (functions of a small number of variables) over the uniform distribution on $\{0,1\}^n$. Approximations of various function classes by juntas is one of the fundamental topics in Boolean function analysis [10], [11], [12], [13] with a growing number of applications in learning theory, computational complexity and algorithms [14], [15], [16], [17], [1]. A classical result in this area is Friedgut's theorem [11] which states that every boolean function f is ϵ-close to a function of $2^{O(\mathsf{Infl}(f)/\epsilon^2)}$ variables, where $\mathsf{Infl}(f)$ is the total influence of f (see Sec. IV-A for the formal definition). Such result is not known for general real-valued functions, and in fact one natural generalization Freidgut's theorem is known not to hold [16]. However, it was recently shown [1] that every submodular function with range $[0,1]$ is ϵ close in ℓ_2-norm to a $2^{O(1/\epsilon^2)}$-junta. Stronger results are known in the special case when a submodular function only takes k different values (for some small k). For this case Blais *et al.* prove existence of a junta of size $(k\log(1/\epsilon))^{O(k)}$ [18] and Feldman *et al.* give $(2^k/\epsilon)^5$ bound [1].

As in [1], our interest in approximation by juntas is motivated by applications to learning of submodular and XOS functions. The question of learning submodular functions from random examples was first formally considered by Balcan and Harvey [2] who motivate it by learning of valuation functions. Reconstruction of submodular 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 *et al.* [19]. These works and widespread applications of submodular functions have recently lead to significant attention to several additional variants of the problem of learning and testing submodular functions as well as their structural properties [20], [21], [22], [23], [24], [25], [1], [18]. We survey related work in more detail in Section

I-B.

A. Our Results

Our work addresses the following two questions: (i) what is the optimal size of junta that ϵ-approximates a submodular function, and in particular whether the known bounds are optimal; (ii) which more general classes of real-valued functions can be approximated by juntas, and in particular whether XOS functions have such approximations.

In short, we provide the following answers: (i) For submodular functions, the optimal ϵ-approximating junta has size $O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$. This is an exponential improvement over the bounds in [1], [18] which shows that submodular functions behave almost as linear functions (which are submodular) and are simpler than XOS functions which require a $2^{\Omega(1/\epsilon)}$-junta to approximate. This result is proved using new techniques. (ii) All functions with range in $[0,1]$ and constant total ℓ_1-influence can be approximated in ℓ_2-norm by a $2^{O(1/\epsilon^2)}$-junta. We show that this captures submodular functions, XOS and even self-bounding functions. This result is a real-valued analogue of Friedgut's theorem and is proved using the same technique.

We now describe these structural results formally and then describe new learning and testing algorithms that rely on them.

1) Structural results: Our main structural result is approximation of submodular functions by juntas.

Theorem 1. *For any $\epsilon \in (0, \frac{1}{2})$ and any submodular function $f : \{0,1\}^n \to [0,1]$, there exists a submodular function $g : \{0,1\}^n \to [0,1]$ depending only on a subset of variables $J \subseteq [n]$, $|J| = O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$, such that $\|f-g\|_2 \le \epsilon$.*

In the special case of submodular functions that take values in $\{0,1,\ldots,k\}$ and ϵ being the disagreement probability our result can be simplified to give a junta of size $O(k\log(k/\epsilon))$. This is an exponential improvement over bounds in both [1] and [18] (see Corollary 15 for a formal statement).

Our proof is based on a new procedure that selects variables that are included in the approximating junta for a submodular function f. We view the hypercube as subsets of $\{1,2,\ldots,n\}$ and refer to $f(S\cup\{i\})-f(S)$ as the marginal value of variable i on set S. Iteratively, we add a variable i if its marginal value is large enough with probability at least $1/2$ taken over sparse random subsets of the variables that are already chosen. One of the key pieces of the proof is the use a "*boosting lemma*[1]" on down-monotone events of Goemans and Vondrak [26]. We use it to show that our criterion for selection of the variables implies that, with very high probability over a random and uniform choice of a subset of selected variables, the marginal value of each of the variables that are excluded is small. The probability of having small marginal value is high enough to apply a union bound over all excluded variables. Bounded marginal values are equivalent to Lipschitzness of the function and allow us to rely on *concentration of Lipschitz submodular functions* to replace the functions of excluded variables by constants. Concentration bounds for submodular functions were first given by Boucheron *et al.* [9] and are also a crucial component of some of the prior works in this area [2], [20], [1].

One application of this procedure allows us to reduce the number of variables from n to $O(\frac{1}{\epsilon^2}\log\frac{n}{\epsilon})$. This process can be repeated until the number of variables becomes $O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$.

Using a more involved argument based on the same ideas we show that monotone submodular functions can with high probability be *multiplicatively* approximated by a junta. Formally, g is an multiplicative (α,ϵ)-approximation to f over a distribution D, if $\Pr_D[f(x) \le g(x) \le \alpha f(x)] \ge 1-\epsilon$. In the PMAC learning model, introduced by Balcan and Harvey [2] a learner has to output a hypothesis that multiplicatively (α,ϵ)-approximates the unknown function. It is a relaxation of the worst case multiplicative approximation used in optimization but is more demanding than the ℓ_1/ℓ_2-approximation that is the main focus of our work. We prove the following in the full version [27]:

Theorem 2. *For every monotone submodular function $f : \{0,1\}^n \to \mathbb{R}_+$ and every $\gamma, \epsilon \in (0,1)$, there is a monotone submodular function $h : \{0,1\}^J \to \mathbb{R}_+$ depending only on a subset of variables $J \subseteq [n], |J| = O(\frac{1}{\gamma^2}\log\frac{1}{\gamma\epsilon}\log\frac{1}{\epsilon})$ such that h is a multiplicative $(1+\gamma,\epsilon)$-approximation of f over the uniform distribution.*

We then show that broader classes of functions such as XOS and self-bounding can also be approximated by juntas, although of an exponentially larger size. We denote by $\mathsf{Infl}^1(f)$ the total ℓ_1-influence of f and by $\mathsf{Infl}^2(f)$ the total ℓ_2^2-influence of f (see Sec. IV-A for definitions). We prove the result via the following generalization of the well-known Friedgut's theorem for boolean functions.

Theorem 3. *Let $f : \{0,1\}^n \to \mathbb{R}$ be any function and $\epsilon > 0$. There exists a function $g : \{0,1\}^n \to \mathbb{R}$ depending only on a subset of variables $J \subseteq [n]$, $|J| = 2^{O(\mathsf{Infl}^2(f)/\epsilon^2)} \cdot (\mathsf{Infl}^1(f))^3/\epsilon^4$ such that $\|f-g\|_2 \le \epsilon$. For a submodular, XOS or self-bounding $f : \{0,1\}^n \to [0,1]$, $\mathsf{Infl}^2(f) \le \mathsf{Infl}^1(f) = O(1)$, giving $|J| = 2^{O(1/\epsilon^2)}$.*

Friedgut's theorem gives approximation by a junta of size $2^{O(\mathsf{Infl}(f)/\epsilon^2)}$ for a boolean f. For a boolean function total influence $\mathsf{Infl}(f)$ (also referred to as average sensitivity) is equal to both $\mathsf{Infl}^1(f)$ and $\mathsf{Infl}^2(f)$ (up to a fixed constant factor). Previously it was observed that Friedgut's theorem is not true if $\mathsf{Infl}^2(f)$ is used in place of $\mathsf{Infl}(f)$ in the statement [16]. However we show that with an additional factor which is just polynomial in $\mathsf{Infl}^1(f)$ one can obtain a generalization. O'Donnell and Servedio [16] generalized the Friedgut's theorem to bounded discretized real-valued functions. They prove a bound of $2^{O(\mathsf{Infl}^2(f)/\epsilon^2)}.\gamma^{-O(1)}$, where γ is the discretization step. This special case is easily implied by our bound. Technically, our proof is a simple refinement of the proof of Friedgut's theorem.

[1] The terminology comes from [26] and has no connection with the notion of boosting in machine learning.

The second component of this result is a simple proof that self-bounding functions (and hence submodular and XOS) have constant total ℓ_1-influence. An immediate implication of this fact alone is that self-bounding functions can be approximated by functions of $O(1/\epsilon^2)$ Fourier degree. For the special case of submodular functions this was proved by Cheraghchi *et al.* also using Fourier analysis, namely, by bounding the noise stability of submodular functions [22]. Our more general proof is also substantially simpler.

We show that this result is almost tight, in the sense that even for XOS functions $2^{\Omega(1/\epsilon)}$ variables are necessary for an ϵ-approximation in ℓ_1 (see Thm. 21). Thus we obtain an almost complete picture, in terms of how many variables are needed to achieve an ϵ-approximation depending on the target function — see Figure 1.

2) Applications: We provide several applications of our structural results to learning and testing. These applications are based on new algorithms as well as standard approaches to learning over the uniform distribution.

For submodular functions our main application is a PMAC learning algorithm over the uniform distribution.

Theorem 4. *There exists an algorithm $\mathcal{A}$ that given $\gamma, \epsilon \in (0,1]$ and access to random and uniform examples of a submodular function $f : \{0,1\}^n \to \mathbb{R}_+$, with probability at least $2/3$, outputs a function h which is a multiplicative $(1+\gamma,\epsilon)$-approximation f (over the uniform distribution). Further, $\mathcal{A}$ runs in time $\tilde{O}(n^2) \cdot 2^{\tilde{O}(1/(\epsilon\gamma)^2)}$ and uses $\log(n) \cdot 2^{\tilde{O}(1/(\epsilon\gamma)^2)}$ examples.*

The main building block of this algorithm is an algorithm that finds an ℓ_2-approximating junta of size $\tilde{O}(1/\epsilon^2)$ whose existence is guaranteed by Theorem 1. The main challenge here is that the criterion for including variables used in the proof of Theorem 1 cannot be (efficiently) evaluated using random examples alone. Instead we give a general algorithm to find a larger approximating junta whenever an approximating junta exists. This algorithm relies only on submodularity of the function and in our case finds a junta of size $\tilde{O}(1/\epsilon^5)$. From there one can easily use brute force to find a $\tilde{O}(1/\epsilon^2)$-junta in time $2^{\tilde{O}(1/\epsilon^2)}$.

We show that using the function g returned by this building block we can partition the domain into $2^{\tilde{O}(1/\epsilon^2)}$ subcubes such that on a constant fraction of those subcubes g gives a multiplicative $(1+\gamma,\epsilon)$ approximation. We then apply the building block recursively for $O(\log(1/\epsilon))$ levels. We remark that our PMAC algorithm does not use the multiplicative approximation by a junta given in Theorem 2 since in this case we do not know how to find an approximating junta from random examples and it only applies to monotone submodular functions.

In addition, the algorithm for finding close-to-optimal ℓ_2-approximating junta allows us to learn properly (by outputting a submodular function) in time $2^{\tilde{O}(1/\epsilon^2)}\text{poly}(n)$. Using a standard transformation we can also test whether the input function is submodular or ϵ-far (in ℓ_1) from submodular, in time $2^{\tilde{O}(1/\epsilon^2)} \cdot \text{poly}(n)$ and using just $2^{\tilde{O}(1/\epsilon^2)} + \text{poly}(1/\epsilon)\log n$ random examples. (Using earlier results, this would have been possible only in time doubly-exponential in ϵ.)

Using the junta and low Fourier degree approximation for self-bounding functions (Theorem 3), we give a PAC learning algorithm for XOS functions.

Theorem 5. *There exists an algorithm $\mathcal{A}$ that given $\epsilon > 0$ and access to random uniform examples of an XOS function $f : \{0,1\}^n \to [0,1]$, with probability at least $2/3$, outputs a function h, such that $\|f-h\|_1 \leq \epsilon$. Further, $\mathcal{A}$ runs in time $2^{O(1/\epsilon^4)}poly(n)$ and uses $2^{O(1/\epsilon^4)}\log n$ random examples.*

In this case the algorithm is fairly standard: we use the fact that XOS functions are monotone and hence their influential variables can be detected from random examples (as for example in [29]). Given the influential variables we can exploit the low Fourier degree approximation to find a hypothesis using ℓ_1 regression over the low degree parities (as done in [1]).

This algorithm naturally extends to any *monotone* real-valued function of low total ℓ_1-influence, of which XOS functions are a special case. Using the algorithm in Theorem 5 we also obtain a PMAC-learning algorithm for XOS functions using the same approach as we used for submodular functions. However the dependence of the running time and sample complexity on $1/\gamma$ and $1/\epsilon$ is doubly-exponential in this case. To our knowledge, this is the first PMAC learning algorithm for XOS functions that can achieve constant approximation factor in polynomial time for all XOS functions.

We give the details of these results and several additional implications of our structural results to agnostic learning and testing in the full version of this work [27].

B. Related Work

Reconstruction of submodular functions up to some multiplicative factor (on every point) from value queries was first considered by Goemans *et al.* [19]. They show a polynomial-time algorithm for reconstructing monotone submodular functions with $\tilde{O}(\sqrt{n})$ factor approximation and prove a nearly matching lower bound. This was extended to the class of all subadditive functions in [23] which studies small-size approximate representations of valuation functions (referred to as *sketches*). Theorem 2 shows that allowing an ϵ error probability (over the uniform distribution) makes it possible to get a multiplicative $(1+\gamma)$-approximation using a $\text{poly}(1/\gamma, \log(1/\epsilon))$-sized sketch. This sketch can be found in polynomial time using value queries.

Balcan and Harvey initiated the study of learning submodular functions from random examples coming from an unknown distribution and introduce the PMAC learning model described above [2]. They give a factor $\sqrt{n}$ PMAC learning algorithm and show an information-theoretic factor-$\sqrt[3]{n}$ inapproximability for submodular functions. Subsequently, Balcan *et al.* gave a distribution-independent PMAC learning algorithm for XOS functions that achieves factor $\tilde{O}(\sqrt{n})$ approximation and showed that this it is essentially optimal [24]. They also give a PMAC learning algorithm in which the number of clauses

Class of functions	junta size lower bound	junta size upper bound
linear	$\Omega(1/\epsilon^2)$ [Folkl., see also [27]]	$O(1/\epsilon^2)$ [Folkl.]
coverage	as above	$O(1/\epsilon^2)$ [28]
submodular	as above	$O(1/\epsilon^2 \cdot \log(1/\epsilon))$ [Thm. 1]
XOS and self-bounding	$2^{\Omega(1/\epsilon)}$ [Thm. 21]	$2^{O(1/\epsilon^2)}$ [Thm. 3]
constant total ℓ_1-influence	$2^{\Omega(1/\epsilon)}$ [11]	$2^{O(1/\epsilon^2)}$ [Thm. 3]
constant total ℓ_2^2-influence	$\Omega(n)$ [16]	n

Fig. 1. Overview of junta results: bounds on the size of a junta achieving an ϵ-approximation in ℓ_2 for a function with range $[0, 1]$.

defining the target XOS function determines the complexity and approximation factor that can be achieved (for polynomial-size XOS functions it implies $O(n^{\beta})$-approximation factor in time $n^{O(1/\beta)}$ for any $\beta > 0$).

The lower bound in [2] also implies hardness of learning of submodular function with ℓ_1(or ℓ_2)-error: it is impossible to learn a submodular function $f : \{0,1\}^n \to [0,1]$ in $\mathrm{poly}(n)$ time within any nontrivial ℓ_1-error over general distributions. 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 the setting of uniform/product distributions which is the focus of this paper.

For product distributions, Balcan and Harvey show that 1-Lipschitz monotone submodular functions of minimum nonzero value at least 1 have concentration properties implying a PMAC algorithm with a multiplicative $(O(\log \frac{1}{\epsilon}), \epsilon)$-approximation [2]. The approximation is by a constant function and the algorithm they give approximates the function by its mean on a small sample. Since a constant is a function of 0 variables, their result can be viewed as an extreme case of approximation by a junta. Our result gives multiplicative $(1 + \gamma, \epsilon)$-approximation for arbitrarily small $\gamma, \epsilon > 0$. The main point of Theorem 2, perhaps surprising, is that the number of required variables grows only polynomially in $1/\gamma$ and logarithmically in $1/\epsilon$.

Learning of submodular functions with additive rather than multiplicative guarantees over the uniform distribution was first considered by Gupta *et al.* who were motivated by applications in private data release [20]. They show that submodular functions can be ϵ-approximated by a collection of $n^{O(1/\epsilon^2)}$ ϵ^2-Lipschitz submodular functions. Concentration properties imply that 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 queries in order to build the collection. Cheraghchi *et al.* use an argument based on noise stability to show that submodular functions can be approximated in ℓ_2 by functions of $O(1/\epsilon^2)$ Fourier degree [22]. This leads to an $n^{O(1/\epsilon^2)}$ learning algorithm which uses only random examples and, in addition, works in the agnostic setting. Most recently, Feldman *et al.* show that the decomposition from [20] can be computed by a low-rank binary decision tree [1]. They then show that this decision tree can then be pruned to obtain depth $O(1/\epsilon^2)$ decision tree that approximates a submodular function. This construction implies approximation by a $2^{O(1/\epsilon^2)}$-junta of Fourier degree $O(1/\epsilon^2)$. They used these structural results to give a PAC learning algorithm running in time $\mathrm{poly}(n) \cdot 2^{O(1/\epsilon^4)}$. Note that our multiplicative $(1+\gamma, \epsilon)$-approximation in this case implies $O(\gamma + \epsilon)$ ℓ_2-error (but ℓ_2-error gives no multiplicative guarantees). In [1] it is also shown that $2^{\Omega(\epsilon^{-2/3})}$ random examples (or even value queries) are necessary to PAC learn monotone submodular functions to ℓ_1-error of ϵ. This implies that our learning algorithms for submodular and XOS functions cannot be substantially improved.

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*) [25]. 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 [20] and obtain a $\mathrm{poly}(n) \cdot k^{O(k \log k/\epsilon)}$-time PAC learning algorithm using value queries. In this special case the results in [1] give approximation of submodular functions by junta of size $\mathrm{poly}(2^k/\epsilon)$ and $\mathrm{poly}(2^k/\epsilon, n)$ PAC learning algorithm from random examples. In an independent work, Blais *et al.* prove existence of a junta of size $(k \log(1/\epsilon))^{O(k)}$ and use it to give an algorithm for testing submodularity using $(k \log(1/\epsilon))^{\tilde{O}(k)}$ value queries [18].

It is interesting to remark that several largely unrelated methods point to approximating junta being of exponential size, namely, pruned decision trees in [1]; Friedgut's theorem based analysis in this work; two Sunflower lemma-style arguments in [18]. However, unexpectedly (at least for the authors), polynomial-size junta suffices.

C. Organization

Following preliminaries in Section II we present the proof of our main structural result (Thm. 1) in Section III. In Section IV we give the proof of Thm. 3 and describe an example of a function that proves tightness of our bound for XOS functions. The rest of the results appear in the full version of this work [27].

II. Preliminaries

First, we define submodular, fractionally subadditive and subadditive functions. These classes are well known in combinatorial optimization and there has been a lot of recent interest in these functions in algorithmic game theory, due to their expressive power as *valuations* of self-interested agents.

Definition 6. *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 $\sum_{i:a \in B_i} \beta_i \geq 1 \; \forall a \in A$.*
- *subadditive, if $f(A \cup B) \leq f(A) + f(B)$ for all $A \subseteq B \subseteq N$.*

Submodular functions are not necessarily nonnegative, but in many applications (especially when considering multiplicative approximations), this is a natural assumption. All our additive approximations are shift-invariant and hence also apply to submodular functions with range $[-1/2, 1/2]$ (and can also be scaled in a straightforward way). Fractionally subadditive functions are nonnegative by definition (by considering $A = B_1, \beta_1 > 1$). Fractionally subadditive functions are known equivalently as XOS functions. This class includes all (nonnegative) monotone submodular functions (but does not contain non-monotone functions).

Next, we introduce *self-bounding functions*. Self-bounding functions were defined by Boucheron, Lugosi and Massart [9] as a unifying class of functions that enjoy strong concentration properties. 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. By $\mathbf{0}$, we denote the all-zeroes vector in $\{0, 1\}^n$ (corresponding to $\emptyset$).

McDiarmid and Reed [30] further generalized the notion of self-bounding functions which we present here.

Definition 7. *For a function $f : \{0, 1\}^n \to \mathbb{R}$ and any $x \in \{0, 1\}^n$, let $\min_{x_i} f(x) = \min\{f(x), f(x \oplus e_i)\}$. f is (a, b)-self-bounding, if for all $x \in \{0, 1\}^n$ and $i \in [n]$,*

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

$$\sum_{i=1}^{n} (f(x) - \min_{x_i} f(x)) \quad \leq \quad af(x) + b. \tag{2}$$

We remark that condition (1) forces self-bounding functions to be 1-Lipschitz. This is not important for our results, but we keep the definition from [9] for consistency with the literature.

The class of (a, b)-self-bounding functions enjoys strong (dimension-free) concentration bounds, with appropriate quantitative adjustments depending on a, b [30]. In this paper, we are primarily concerned with $(a, 0)$-self-bounding functions, to which we also refer as a-self-bounding functions. Note that the definition implies that $f(x) \geq 0$ for every a-self-bounding function. Self-bounding functions include (1-Lipschitz) fractionally subadditive functions. To subsume 1-Lipschitz nonmonotone submodular functions, it is sufficient to consider the slightly more general 2-self-bounding functions - see [31]. The 1-Lipschitz condition will not play a role in this paper, as we normalize functions to have values in the $[0, 1]$ range.

The ℓ_1 and ℓ_2-norms of a $f : \{0, 1\}^n \to \mathbb{R}$ are defined by $||f||_1 = \mathbf{E}_{x \sim \mathcal{U}}[|f(x)|]$ and $||f||_2 = (\mathbf{E}_{x \sim \mathcal{U}}[f(x)^2])^{1/2}$, respectively, where $\mathcal{U}$ is the uniform distribution.

Definition 8 (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 x with i-th coordinate set to b. For a function $f : \{0, 1\}^n \to \mathbb{R}$ and index $i \in [n]$ we define $\partial_i f(x) = f(x_{i \leftarrow 1}) - f(x_{i \leftarrow 0})$. We also define $\partial_{i,j} f(x) = \partial_i \partial_j f(x)$.*

A function is monotone (non-decreasing) if and only if for all $i \in [n]$ and $x \in \{0, 1\}^n$, $\partial_i f(x) \geq 0$. For a submodular function, $\partial_{i,j} f(x) \leq 0$, by considering the submodularity condition for $x_{i \leftarrow 0, j \leftarrow 0}$, $x_{i \leftarrow 0, j \leftarrow 1}$, $x_{i \leftarrow 1, j \leftarrow 0}$, and $x_{i \leftarrow 1, j \leftarrow 1}$.

III. Junta Approximations of Submodular Functions

Here we prove Theorem 1, a bound of $\tilde{O}(1/\epsilon^2)$ on the size of a junta needed to approximate a submodular function bounded by $[0, 1]$ within an additive error of ϵ. The core of our proof is the following (seemingly weaker) statement. We remark that logarithms in this paper are base 2.

Lemma 9. *For any $\epsilon \in (0, \frac{1}{2})$ and any submodular function $f : \{0, 1\}^J \to [0, 1]$, there exists a submodular function $h : \{0, 1\}^J \to [0, 1]$ depending only on a subset of variables $J' \subseteq J$, $|J'| \leq \frac{128}{\epsilon^2} \log \frac{16|J|}{\epsilon^2}$, such that $\|f - h\|_2 \leq \frac{1}{2}\epsilon$.*

Note that if $|J| = n$ and $\epsilon = \Omega(1)$, Lemma 9 reduces the number of variables to $O(\log n)$ rather than a constant. However, we show that this is enough to prove Theorem 1, effectively by repeating this argument. In fact, it was previously shown [1] that submodular functions can be ϵ-approximated by functions of $2^{O(1/\epsilon^2)}$ variables. One application of Lemma 9 to this result brings the number of variables down to $\tilde{O}(\frac{1}{\epsilon^4})$, and another repetition of the same argument brings it down to $O(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon})$. This is a possible way to prove Theorem 1. Nevertheless, we do not need to rely on this previous result, and we can easily derive Theorem 1 directly from Lemma 9 (see full version for the details). In the rest of this section, our goal is to prove Lemma 9.

What we need. Our proof relies on two previously known facts: a concentration result for submodular functions, and a "boosting lemma" for down-monotone events.

Concentration of submodular functions. It is known that a 1-Lipschitz nonnegative submodular function f is concentrated within a standard deviation of $O(\sqrt{\mathbf{E}[f]})$ [9], [31]. This fact was also used in previous work on learning of submodular functions [2], [20], [1]. Exponential tail bounds are known in this case, but we do not even need this. We quote the following result which follows from the Efron-Stein inequality (see [1] for a proof).

Lemma 10. *For any α-Lipschitz monotone submodular function $f : \{0, 1\}^n \to \mathbb{R}_+$,*

$$\mathbf{Var}[f] \leq \alpha \, \mathbf{E}[f].$$

For any α-Lipschitz (nonmonotone) submodular function $f : \{0, 1\}^n \to \mathbb{R}_+$,

$$\mathbf{Var}[f] \leq 2\alpha \, \mathbf{E}[f].$$

Boosting lemma for down-monotone events. The following was proved as Lemma 3 in [26].

Lemma 11. *Let $\mathcal{F} \subseteq \{0,1\}^X$ be down-monotone (if $x \in \mathcal{F}$ and $y \leq x$ coordinate-wise, then $y \in \mathcal{F}$). For $p \in (0,1)$, define*

$$\sigma_p = \Pr[X(p) \in \mathcal{F}]$$

where $X(p)$ is a random subset of X, each element sampled independently with probability p. Then

$$\sigma_p = (1-p)^{\phi(p)}$$

where $\phi(p)$ is a non-decreasing function for $p \in (0,1)$.

a) The proof of Lemma 9: Given a submodular function $f : \{0,1\}^J \to [0,1]$, let $F : [0,1]^J \to [0,1]$ denote the multilinear extension of f: $F(x) = \mathbf{E}[f(\hat{x})]$ where $\hat{x}$ has independently random 0/1 coordinates with expectations x_i. We also denote by $\mathbf{1}_S$ the characteristic vector of a set S.

Algorithm 12. *Given $f : \{0,1\}^J \to [0,1]$, produce a small set of important coordinates J' as follows (for parameters $\alpha, \delta > 0$):*

- *Set $S = T = \emptyset$.*
- *As long as there is $i \notin S$ such that* $\Pr[\partial_i f(\mathbf{1}_{S(\delta)}) > \alpha] > 1/2$, *include i in S.*
  (This step is sufficient for monotone submodular functions.)
- *As long as there is $i \notin T$ such that* $\Pr[\partial_i f(\mathbf{1}_{J \setminus T(\delta)}) < -\alpha] > 1/2$, *include i in T.*
  (This step deals with non-monotone submodular functions.)
- *Return $J' = S \cup T$.*

The intuition here (for monotone functions) is that we include greedily all variables whose contribution is significant, when measured at a random point where the variables chosen so far are set to 1 with a (small) probability δ. The reason for this is that we can bound the number of such variables, and at the same time we can prove that the contribution of unchosen variables is very small *with high probability*, when the variables in J' are assigned uniformly at random (this part uses the boosting lemma). This is helpful in estimating the approximation error of this procedure.

First, we bound the number of variables chosen by the procedure. The argument is essentially that if the procedure had selected too many variables, their expected cumulative contribution would exceed the bounded range of the function. This argument would suffice for monotone submodular functions. The final proof is somewhat technical because of the need to deal with potentially negative discrete derivatives of non-monotone submodular functions.

Lemma 13. *The number of variables chosen by the procedure above is $|J'| \leq \frac{4}{\alpha\delta}$.*

Proof. For each $i \in S$, let $S_{<i}$ be the subset of variables in S included before the selection of i. For a set $R \subseteq S$ let $R_{<i}$ denote $R \cap S_{<i}$. Further, for $R \subseteq S$, let us define R^+ to be the set where $i \in R^+$ iff $i \in R$ and $\partial_i f(\mathbf{1}_{R_{<i}}) > \alpha$; in other words, these are all the elements in R that have a marginal contribution more than α to the previously included elements.

For each variable i included in S, we have by definition $\Pr[\partial_i f(\mathbf{1}_{S_{<i}(\delta)}) > \alpha] > 1/2$. Since each $i \in S$ appears in $S(\delta)$ with probability δ, and (independently) $\partial_i f(\mathbf{1}_{S_{<i}(\delta)}) > \alpha$ with probability at least $1/2$, we get that each element of S appears in $S(\delta)^+$ with probability at least $\delta/2$. In expectation, $\mathbf{E}[|S(\delta)^+|] \geq \frac{1}{2}\delta|S|$. Also, for any set $R \subseteq S$ and each $i \in R^+$, submodularity implies that $\partial_i f(\mathbf{1}_{R^+_{<i}}) \geq \partial_i f(\mathbf{1}_{S_{<i}}) > \alpha$, since $R^+_{<i} \subseteq R_{<i} \subseteq S_{<i}$. Now we get that

$$f(R^+) = f(\mathbf{0}) + \sum_{i \in R^+} \partial_i f(\mathbf{1}_{R^+_{<i}}) > \alpha|R^+|.$$

From here we obtain that

$$\mathbf{E}[f(S(\delta)^+)] > \alpha\,\mathbf{E}[|S(\delta)^+|] \geq \frac{1}{2}\alpha\delta|S|.$$

This implies that $|S| \leq \frac{2}{\alpha\delta}$, otherwise the expectation would exceed the range of f, which is $[0,1]$.

To bound the size of T we observe that the function $\bar{f}$ defined as $\bar{f}(\mathbf{1}_R) = f(\mathbf{1}_{J \setminus R})$ for every $R \subseteq J$ is submodular and for every $i \in J$, $\partial_i \bar{f}(\mathbf{1}_R) = -\partial_i f(\mathbf{1}_{J \setminus R})$. The criterion for including the variables in T is the same as criterion of including the variables in S used for function $\bar{f}$ in place of f. Therefore, by an analogous argument, we cannot include more than $\frac{2}{\alpha\delta}$ elements in T, hence $|J'| = |S \cup T| \leq \frac{4}{\alpha\delta}$. □

The next step in the analysis replaces the condition used by Algorithm 12 by a probability bound exponentially small in $1/\delta$. The tool that we use here is the "boosting lemma" (Lemma 11) which amplifies the probability bound from $1/2$ to $1/2^{1/(2\delta)}$, as the sampling probability goes from δ to $1/2$.

Lemma 14. *With the same notation as above, if $\delta \leq 1/2$, then for any $i \in J \setminus J'$*

$$\Pr[\partial_i f(\mathbf{1}_{J'(1/2)}) > \alpha] \leq 2^{-1/(2\delta)}$$

and

$$\Pr[\partial_i f(\mathbf{1}_{J \setminus J'(1/2)}) < -\alpha] \leq 2^{-1/(2\delta)}.$$

Proof. Let us prove the first inequality; the second one will be similar. First, we know by the selection rule of the algorithm that for any $i \notin J'$,

$$\Pr[\partial_i f(\mathbf{1}_{S(\delta)}) > \alpha] \leq 1/2.$$

By submodularity of f we get that for any $i \notin J'$,

$$\Pr[\partial_i f(\mathbf{1}_{J'(\delta)}) > \alpha] \leq 1/2.$$

Denote by $\mathcal{F} \subseteq \{0,1\}^{J'}$ the family of points x such that $\partial_i f(x) > \alpha$. By the submodularity of f, which is equivalent to partial derivatives being non-increasing, $\mathcal{F}$ is a down-monotone set: if $y \leq x \in \mathcal{F}$, then $y \in \mathcal{F}$. If we define $\sigma_p = \Pr[J'(p) \in \mathcal{F}]$ as in Lemma 11, we have $\sigma_\delta \leq 1/2$. Therefore, by Lemma 11, $\sigma_p = (1-p)^{\phi(p)}$ where $\phi(p)$ is a non-decreasing function. For $p = \delta$, we get $\sigma_\delta = (1-\delta)^{\phi(\delta)} \leq 1/2$, which implies $\phi(\delta) \geq 1/(2\delta)$

(note that $(1-\delta)^{1/(2\delta)} \geq 1/2$ for any $\delta \in [0,1/2]$). As $\phi(p)$ is non-decreasing, we must also have $\phi(1/2) \geq 1/(2\delta)$. This means $\sigma_{1/2} = (1/2)^{\phi(1/2)} \leq 1/2^{1/(2\delta)}$. Recall that $\sigma_{1/2} = \Pr[J'(1/2) \in \mathcal{F}] = \Pr[\partial_i f(\mathbf{1}_{J'(p)}) > \alpha]$ so this proves the first inequality.

For the second inequality, we denote similarly $\mathcal{F} = \{F \subseteq J' : \partial_i f(\mathbf{1}_{J \setminus F}) < -\alpha\}$. Again, this is a down-monotone set by the submodularity of f. By the selection rule of the algorithm, $\sigma_\delta = \Pr[J'(\delta) \in \mathcal{F}] = \Pr[\partial_i f(\mathbf{1}_{J \setminus J'(\delta)}) < -\alpha] \leq \Pr[\partial_i f(\mathbf{1}_{J \setminus T(\delta)}) < -\alpha] \leq 1/2$. This implies by Lemma 11 that $\sigma_{1/2} = \Pr[J'(1/2) \in \mathcal{F}] \leq 1/2^{1/(2\delta)}$. This proves the second inequality. □

Proof of Lemma 9. Given a submodular function $f : \{0,1\}^J \to [0,1]$, we construct a set of coordinates $J' \subseteq J$ as described above, with parameters $\alpha = \frac{1}{16}\epsilon^2$ and $\delta = 1/(2\log\frac{16|J|}{\epsilon^2})$. Lemma 13 guarantees that $|J'| \leq \frac{4}{\alpha\delta} = \frac{128}{\epsilon^2}\log\frac{16|J|}{\epsilon^2}$.

Let us use $x_{J'}$ to denote the $|J'|$-tuple of coordinates of x indexed by J'. Consider the subcube of $\{0,1\}^J$ where the coordinates on J' are fixed to be $x_{J'}$. In the following, all expectations are over a uniform distribution on the respective subcube, unless otherwise indicated. We denote by $f_{x_{J'}}$ the restriction of f to this subcube, $f_{x_{J'}}(x_{\bar{J'}}) = f(x_{J'}, x_{\bar{J'}})$. We define $h : \{0,1\}^J \to [0,1]$ to be the function obtained by replacing each $f_{x_{J'}}$ by its expectation over the respective subcube:

$$h(x) = \mathbf{E}[f_{x_{J'}}] = \mathop{\mathbf{E}}_{y_{\bar{J'}}}[f(x_{J'}, y_{\bar{J'}})].$$

Obviously h depends only on the variables in J' and it is easy to see that it is submodular with range in $[0,1]$. It remains to estimate the distance of h from f. Observe that

$$\begin{aligned}
\|f-h\|_2^2 &= \mathop{\mathbf{E}}_{x}[(f(x)-h(x))^2] \\
&= \mathop{\mathbf{E}}_{x_{J'}}\mathop{\mathbf{E}}_{y_{\bar{J'}}}[(f(x_{J'}, y_{\bar{J'}}) - h(x_{J'}, y_{\bar{J'}}))^2] \\
&= \mathop{\mathbf{E}}_{x_{J'}}\mathop{\mathbf{E}}_{y_{\bar{J'}}}[(f_{x_{J'}}(y_{\bar{J'}}) - \mathbf{E}[f_{x_{J'}}])^2] \\
&= \mathop{\mathbf{E}}_{x_{J'}}[\mathbf{Var}[f_{x_{J'}}]].
\end{aligned}$$

We partition the points $x_{J'} \in \{0,1\}^{J'}$ into two classes:

1) Call $x_{J'}$ bad, if there is $i \in J \setminus J'$ such that
   - $\partial_i f(x_{J'}) > \alpha$, or
   - $\partial_i f(x_{J'} + \mathbf{1}_{J \setminus J'}) < -\alpha$.

   In particular, we call $x_{J'}$ bad for the coordinate i where this happens.
2) Call $x_{J'}$ good otherwise, i.e. for every $i \in J \setminus J'$ we have
   - $\partial_i f(x_{J'}) \leq \alpha$, and
   - $\partial_i f(x_{J'} + \mathbf{1}_{J \setminus J'}) \geq -\alpha$.

Consider a good point $x_{J'}$ and the restriction of f to the respective subcube, $f_{x_{J'}}$. The condition above means that for every $i \in J \setminus J'$, the marginal value of i is at most α at the bottom of this subcube, and at least $-\alpha$ at the top of this subcube. By submodularity, it means that the marginal values are between $[-\alpha, \alpha]$, for all points of this subcube. Hence, $f_{x_{J'}}$ is a α-Lipschitz submodular function. By Lemma 10,

$$\mathbf{Var}[f_{x_{J'}}] \leq 2\alpha\,\mathbf{E}[f_{x_{J'}}] \leq \frac{1}{8}\epsilon^2$$

considering that $\alpha = \frac{1}{16}\epsilon^2$ and $f_{x_{J'}}$ has values in $[0,1]$.

If $x_{J'}$ is bad, then we do not have a good bound on the variance of $f_{x_{J'}}$. However, there cannot be too many bad points $x_{J'}$, due to Lemma 14: Observe that the distribution of $x_{J'}$, uniform in $\{0,1\}^{J'}$, is the same as what we denoted by $\mathbf{1}_{J'(1/2)}$ in Lemma 14, and the distribution of $x_{J'} + \mathbf{1}_{J \setminus J'}$ is the same as $\mathbf{1}_{J \setminus J'(1/2)}$. By Lemma 14, we have that for each $i \in J \setminus J'$, the probability that $x_{J'}$ is bad for i is at most $2 \cdot 2^{1/(2\delta)} = \frac{\epsilon^2}{8|J|}$. By a union bound over all coordinates $i \in J \setminus J'$, the probability that $x_{J'}$ is bad is at most $\frac{1}{8}\epsilon^2$.

Now we can estimate the ℓ_2-distance between f and h:

$$\begin{aligned}
\|f-h\|_2^2 &= \mathop{\mathbf{E}}_{x_{J'} \in \{0,1\}^{J'}}[\mathbf{Var}[f_{x_{J'}}]] \\
&\leq \Pr[x_{J'} \text{ is bad}] \cdot 1 + \Pr[x_{J'} \text{ is good}] \cdot \\
&\quad \cdot \mathop{\mathbf{E}}_{\text{good } x_{J'} \in \{0,1\}^{J'}}[\mathbf{Var}[f_{x_{J'}}]] \\
&\leq \Pr[x_{J'} \text{ is bad}] + \max_{\text{good } x_{J'} \in \{0,1\}^{J'}}[\mathbf{Var}[f_{x_{J'}}]] \\
&\leq \frac{1}{8}\epsilon^2 + \frac{1}{8}\epsilon^2 = \frac{1}{4}\epsilon^2.
\end{aligned}$$

Hence, we conclude that $\|f-h\|_2 \leq \frac{1}{2}\epsilon$ as desired. □

We now briefly examine the special case of a submodular function taking values in $\{0, \frac{1}{k}, \frac{2}{k}, \ldots, 1\}$ for some integer k. This is just a scaled version of the pseudo-boolean case considered in [25] and [18]. By choosing $\alpha = \frac{1}{k+1}$ and $\delta = 1/\log(2|J|/\epsilon)$ in the proof above we will obtain that α-Lipschitz function must be a constant (and, in particular, independent of all the variables in $J \setminus J'$). This means that we obtain exact equality for all but the "bad" values of $x_{J'}$. The fraction of such values is at most $2 \cdot 2^{1/\delta} \cdot |J| \leq \epsilon$ and therefore the submodular function $h(x) = f(x_J, \mathbf{1}_{J \setminus J'})$ equals f with probability at least $1-\epsilon$. As before, after one application we get a $O(k \cdot \log(n/\epsilon))$-junta and by repeating the application we can obtain a $O(k \cdot \log(k/\epsilon))$-junta.

Corollary 15. *For any integer $k \geq 1$, $\epsilon \in (0, \frac{1}{2})$ and any submodular function $f : \{0,1\}^n \to \{0,1,\ldots,k\}$, there exists a submodular function $g : \{0,1\}^n \to \{0,1,\ldots,k\}$ depending only on a subset of variables $J \subseteq [n]$, $|J| = O(k\log\frac{k}{\epsilon})$, such that $\Pr_{\mathcal{U}}[f \neq g] \leq \epsilon$.*

IV. Approximation of Low-Influence Functions by Juntas

Here we show how structural results for submodular (weaker than the one in Section III), XOS and self-bounding functions can be proved in a unified manner using the notion of total influence.

A. Preliminaries: Fourier Analysis

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. The *spectral* ℓ_1*-norm* of f is defined as $||\hat{f}||_1 = \sum_{S \subseteq [n]} |\hat{f}(S)|$.

Observe that $\partial_i f(x) = 2\sum_{S \ni i} \hat{f}(S)\chi_{S\setminus\{i\}}(x)$, and $\partial_{i,j} f(x) = 4\sum_{S \ni i,j} \hat{f}(S)\chi_{S\setminus\{i,j\}}(x)$.

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. [32], [33]).

Definition 16 (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 $\mathsf{Infl}_i^\kappa(f) = \|\frac{1}{2}\partial_i f\|_\kappa^\kappa = \mathbf{E}[|\frac{1}{2}\partial_i f|^\kappa]$. We define $\mathsf{Infl}^\kappa(f) = \sum_{i \in [n]} \mathsf{Infl}_i^\kappa(f)$ and refer to it as the* total ℓ_κ^κ-influence *of f. For a boolean function $f : \{0,1\}^n \to \{0,1\}$, $\mathsf{Infl}(f)$ is defined as $2\mathsf{Infl}^1(f)$ and is also referred to as* average sensitivity.

The most commonly used notion of influence for real-valued functions is the ℓ_2^2-influence which satisfies

$$\mathsf{Infl}_i^2(f) = \left\|\frac{1}{2}\partial_i f\right\|_2^2 = \sum_{S \ni i} \hat{f}^2(S) \ .$$

From here, the total ℓ_2^2-influence is equal to $\mathsf{Infl}^2(f) = \sum_S |S|\hat{f}^2(S)$.

B. Self-bounding Functions Have Low Total Influence

A key fact that we prove is that submodular, XOS and self-bounding functions have low total ℓ_1-influence.

Lemma 17. *Let $f : \{0,1\}^n \to \mathbb{R}_+$ be an a-self-bounding function. Then $\mathsf{Infl}^1(f) \leq a \cdot \|f\|_1$. In particular, for XOS $f : \{0,1\}^n \to [0,1]$, $\mathsf{Infl}^1(f) \leq 1$. For a submodular $f : \{0,1\}^n \to [0,1]$, $\mathsf{Infl}^1(f) \leq 2$.*

Proof. We have

$$\begin{aligned}\mathsf{Infl}^1(f) &= \frac{1}{2}\sum_{i=1}^n \mathbf{E}[|f(x_{i\leftarrow 1}) - f(x_{i\leftarrow 0})|] \\ &= \sum_{i=1}^n \mathbf{E}[(f(x) - f(x \oplus e_i))_+]\end{aligned}$$

where $x \oplus e_i$ is x with the i-th bit flipped, and $(\bullet)_+ = \max\{\bullet, 0\}$ is the positive part of a number. (Note that each difference $|f(x_{i\leftarrow 1}) - f(x_{i\leftarrow 0})|$ is counted twice in the first expectation and once in the second expectation.) By using the property of a-self-bounding functions, we know that $\sum_{i=1}^n (f(x) - f(x \oplus e_i))_+ \leq af(x)$, which implies

$$\mathsf{Infl}^1(f) = \sum_{i=1}^n \mathbf{E}[(f(x) - f(x \oplus e_i))_+] \leq a\,\mathbf{E}[|f(x)|] = a\|f\|_1.$$

Finally, we recall that an XOS function is self-bounding and a non-negative submodular function is 2-self-bounding (see [31]). □

We note that for functions with a $[0,1]$ range, $\mathsf{Infl}^2(f) \leq \mathsf{Infl}^1(f)$, hence the above lemma also gives a bound on $\mathsf{Infl}^2(f)$. It is well-known that functions of low total ℓ_2^2-influence can be approximated by low-degree polynomials. We recap this fact here.

Lemma 18. *Let $f : \{0,1\}^n \to \mathbb{R}$ be any function and let d be any positive integer. Then $\sum_{S \subseteq [n], |S| > d} \hat{f}(S)^2 \leq \mathsf{Infl}^2(f)/d$.*

Proof. From the definition of $\mathsf{Infl}_i^2(f)$, we get that $\mathsf{Infl}^2(f) = \sum_{S \subseteq [n]} |S|\hat{f}(S)^2$. Hence

$$\sum_{S \subseteq [n],\ |S| > d} \hat{f}(S)^2 \leq \frac{1}{d}\mathsf{Infl}^2(f) \ .$$

□

This gives a simple proof that submodular and XOS functions are ϵ-approximated in ℓ_2 by polynomials of degree $2/\epsilon^2$ (which was proved for submodular functions in [22]). We next show a stronger statement, that these functions are ϵ-approximated by $2^{O(1/\epsilon^2)}$-juntas of Fourier degree $O(1/\epsilon^2)$.

C. Friedgut's Theorem for Real-Valued Functions

As we have shown in Lemma 17, self-bounding functions have low total ℓ_1-influence. A celebrated result of Friedgut [11] shows that any Boolean function on $\{0,1\}^n$ of low total influence is close to a function that depends on few variables. It is therefore natural to try and apply Friedgut's result to our setting. A commonly considered generalization of Boolean influences to real-valued functions uses ℓ_2^2-influences which can be easily expressed using Fourier coefficients (e.g. [34]). However Friedgut-style result is not true for real-valued functions when ℓ_2^2-influences are used [16], [27]. This issue also arises in the problem of learning real-valued monotone decision trees by O'Donnell and Servedio [16]. They overcome the problem by first discretizing the function and proving that Friedgut's theorem can be extended to the discrete case (as long as the discretization step is not too small). The problem with using this approach for submodular functions is that it does not preserve submodularity and can increase total influence of the resulting function to $\Omega(\sqrt{n})$ with discretization parameter necessary for the approach to work (consider for example a linear function $\frac{1}{n}\sum_i x_i$).

Here we instead prove a generalization of Friedgut's theorem to all real-valued functions. We show that Friedgut's theorem would hold for real-valued functions if the total ℓ_κ^κ-influence is small in addition to total ℓ_2^2-influence for any constant $\kappa \in [1,2)$. Self-bounding functions have low total

ℓ_1-influence and hence for our purposes $\kappa = 1$ would suffice. We prove the slightly more general version as it could be useful elsewhere (and the proof is essentially the same).

Theorem 19. *Let $f : \{0,1\}^n \to \mathbb{R}$ be any function and $\epsilon, \kappa \in (0,1)$. Let $d = 2 \cdot \mathsf{Infl}^2(f)/\epsilon$ and let*

$$I = \{i \in [n] \mid \mathsf{Infl}_i^{\kappa}(f) \geq \alpha\} \textit{ for}$$
$$\alpha = \left((\kappa - 1)^{d-1} \cdot \epsilon/(2\mathsf{Infl}^{\kappa}(f))\right)^{\kappa/(2-\kappa)} .$$

Then for the set $\mathcal{I}_d = \{S \subseteq I \mid |S| \leq d\}$ we have $\sum_{S \notin \mathcal{I}_d} \hat{f}(S)^2 \leq \epsilon$.

To obtain Theorem 3 from this statement we use it with ϵ^2 error and note that $g = \sum_{S \in \mathcal{I}_d} \hat{f}(S)\chi_S$ is a function of Fourier degree d that depends only on variables in I. Further, $\|f - g\|_2^2 \leq \epsilon^2$ and the set I has size of at most

$$|I| \leq \mathsf{Infl}^{\kappa}(f)/\alpha = 2^{O(\mathsf{Infl}^2(f)/\epsilon^2)} \cdot \epsilon^{2\kappa/(2-\kappa)} \cdot (\mathsf{Infl}^{\kappa}(f))^{2/(2-\kappa)}. \tag{3}$$

Also note that Theorem 19 does not allow us to directly bound $|I|$ in terms of $\mathsf{Infl}^1(f)$ since it does not apply to $\kappa = 1$. However for every $\kappa \in [1,2]$, $\mathsf{Infl}^{\kappa}(f) \leq \mathsf{Infl}^1(f) + \mathsf{Infl}^2(f)$ and therefore we can also bound $|I|$ using equation (3) for $\kappa = 4/3$ and then substituting $\mathsf{Infl}^{4/3}(f) \leq \mathsf{Infl}^1(f) + \mathsf{Infl}^2(f)$. This gives the proof of Theorem 3 (first part). The second part of Theorem 3 now follows from Lemma 17.

Our proof of Theorem 19 is a simple modification of the proof of Friedgut's theorem from [35] and can be found in the full version of the work [27]. For functions that have low total ℓ_1-influence we also easily obtain the following corollary of Th. 19.

Corollary 20. *Let $f : \{0,1\}^n \to [0,1]$ be any function and $\epsilon > 0$. For $d = 2 \cdot \mathsf{Infl}^1(f)/\epsilon^2$ and $\alpha = 2^{-4d}$ let*

$$I = \{i \in [n] \mid \mathsf{Infl}_i^1(f) \geq \alpha\}.$$

There exists a function p of Fourier degree d over variables in I, such that $\|f - p\|_2 \leq \epsilon$ and $\|\hat{p}\|_1 \leq 2^{O(\mathsf{Infl}^1(f)^2/\epsilon^4)}$.

D. Lower Bound On Junta Size For XOS Functions

Here we prove that Theorem 3 is close-to-tight and, in particular, Theorem 1 cannot be extended to XOS functions. In fact, we show that $2^{\Omega(1/\epsilon)}$ variables are necessary for an ϵ-approximation to an XOS function. Our lower bound is based on the Tribes DNF function studied by Ben-Or and Linial [32] with AND replaced by a linear function. The Tribes DNF was also used by Friedgut to prove tightness of his theorem for boolean functions [11].

Theorem 21. *Suppose that $n = ab$ where $b = 2^a$ and consider an XOS function*

$$f(x) = \frac{1}{a} \max_{1 \leq j \leq b} \sum_{i \in A_j} x_i$$

where $(A_1, \ldots, A_b)$ is a partition of $[n]$ into sets of size $|A_j| = a$. Then every function $g : \{0,1\}^n \to \mathbb{R}$ that depends on fewer than 2^{a-1} variables has $\|f - g\|_1 = \Omega(1/a)$.

Acknowledgements

We would like to thank Seshadhri Comandur, Pravesh Kothari and the anonymous FOCS referees for their comments and useful suggestions.

References

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

[2] M. Balcan and N. Harvey, "Submodular functions: Learnability, structure, and optimization," *CoRR*, vol. abs/1008.2159, 2012, earlier version in STOC 2011.

[3] J. Edmonds, "Matroids, submodular functions and certain polyhedra," *Combinatorial Structures and Their Application*.

[4] L. Lovász, "Submodular functions and convexity," *Mathematical Programmming: The State of the Art*, pp. 235–257, 1983.

[5] A. Frank, "Matroids and submodular functions," *Annotated Biblographies in Combinatorial Optimization*, pp. 65–80, 1997.

[6] C. Guestrin, A. Krause, and A. Singh, "Near-optimal sensor placements in gaussian processes," in *ICML*, 2005, pp. 265–272.

[7] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: maximizing information while minimizing communication cost," in *IPSN*, 2006, pp. 2–10.

[8] D. J. L. B. Lehmann and N. Nisan, "Combinatorial auctions with decreasing marginal utilities," *Games and Economic Behavior*, vol. 55, pp. 1884–1899, 2006.

[9] S. Boucheron, G. Lugosi, and P. Massart, "A sharp concentration inequality with applications," *Random Struct. Algorithms*, vol. 16, no. 3, pp. 277–292, 2000.

[10] N. Nisan and M. Szegedy, "On the degree of boolean functions as real polynomials," *Computational Complexity*, vol. 4, pp. 462–467, 1992.

[11] E. Friedgut, "Boolean functions with low average sensitivity depend on few coordinates," *Combinatorica*, vol. 18, no. 1, pp. 27–35, 1998.

[12] J. Bourgain, "On the distribution of the fourier spectrum of boolean functions," *Israel Journal of Mathematics*, vol. 131(1), pp. 269–276, 2002.

[13] E. Friedgut, G. Kalai, and A. Naor, "Boolean functions whose Fourier transform is concentrated on the first two levels," *Adv. in Appl. Math*, vol. 29, 2002.

[14] I. Dinur and S. Safra, "On the hardness of approximating minimum vertex cover," *Annals of Mathematics*, vol. 162, 2005.

[15] R. Krauthgamer and Y. Rabani, "Improved lower bounds for embeddings into L1," in *SODA*, 2006, pp. 1010–1017.

[16] R. O'Donnell and R. Servedio, "Learning monotone decision trees in polynomial time," *SIAM J. Comput.*, vol. 37, no. 3, pp. 827–844, 2007.

[17] P. Gopalan, R. Meka, and O. Reingold, "DNF sparsification and a faster deterministic counting algorithm," in *CCC*, 2012, pp. 126–135.

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

[19] M. Goemans, N. Harvey, S. Iwata, and V. Mirrokni, "Approximating submodular functions everywhere," in *SODA*, 2009, pp. 535–544.

[20] A. Gupta, M. Hardt, A. Roth, and J. Ullman, "Privately releasing conjunctions and the statistical query barrier," in *STOC*, 2011, pp. 803–812.

[21] C. Seshadhri and J. Vondrák, "Is submodularity testable?" in *Innovations in computer science*, 2011, pp. 195–210.

[22] M. Cheraghchi, A. Klivans, P. Kothari, and H. Lee, "Submodular functions are noise stable," in *SODA*, 2012, pp. 1586–1592.

[23] A. Badanidiyuru, S. Dobzinski, H. Fu, R. Kleinberg, N. Nisan, and T. Roughgarden, "Sketching valuation functions," in *SODA*, 2012, pp. 1025–1035.

[24] M. Balcan, F. Constantin, S. Iwata, and L. Wang, "Learning valuation functions," *COLT*, vol. 23, pp. 4.1–4.24, 2012.

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

[26] M. Goemans and J. Vondrák, "Covering minimum spanning trees of random subgraphs," *Random Struct. Algorithms*, vol. 29, no. 3, pp. 257–276, 2006.

[27] V. Feldman and J. Vondrák, "Optimal bounds on approximation of submodular and xos functions by juntas," *CoRR*, vol. abs/1307.3301, 2013.

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

[29] R. Servedio, "On learning monotone DNF under product distributions," *Information and Computation*, vol. 193, no. 1, pp. 57–74, 2004.

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

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

[32] M. Ben-Or and N. Linial, "Collective coin flipping, robust voting schemes and minima of banzhaf values," in *FOCS*, 1985, pp. 408–416.

[33] J. Kahn, G. Kalai, and N. Linial, "The influence of variables on Boolean functions," in *FOCS*, 1988, pp. 68–80.

[34] I. Dinur, E. Friedgut, G. Kindler, and R. O'Donnell, "On the Fourier tails of bounded functions over the discrete cube," in *STOC*, 2006, pp. 437–446.

[35] I. Dinur and E. Friedgut, "Lecture notes for analytical methods in combinatorics and computer-science (lect 5)," 2005, available at `http://www.cs.huji.ac.il/˜analyt/`.