Optimal Bounds on Approximation of Submodular and XOS Functions by Juntas

Vitaly Feldman
IBM Research - Almaden

Jan Vondrák
IBM Research - Almaden

April 27, 2015

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 [FKV13]. 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 generalization 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 [BH12] 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 bounds for approximation by juntas. As follows from the lower bounds in [FKV13] both of these algorithms are close to optimal. We also give applications for proper learning, testing and agnostic learning of these classes.

1 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 [Edm70, Lov83, Que95, Fra97, FFI01]. 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 [GKS05, KGGK06, KSG08] 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 [BLN06, DS06, Von08]. Along with submodular functions, other related classes have been studied in the 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 [BLM00]. We refer the reader to Section 2 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 [NS92, Fri98, Bou02, FKN02] with a growing number of applications in learning theory, computational complexity and algorithms [DS05, CKK$^+$06, KR06, OS07, KR08, GMR12, FKV13]. A classical result in this area is Friedgut's theorem [Fri98] 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. 4.1 for the formal definition). Such a result is not known for general real-valued functions, and in fact one natural generalization Freidgut's theorem is known not to hold [OS07]. However, it was recently shown [FKV13] 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 $(k\log(1/\epsilon))^{O(k)}$ [BOSY13] and Feldman *et al.* give a $(2^k/\epsilon)^5$ bound [FKV13].

As in [FKV13], 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 [BH12] who motivate it by learning of valuation functions. Reconstruction of submodular 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 *et al.* [GHIM09]. These works and wide-spread 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 [GHRU11, SV11, CKKL12, BDF$^+$12, BCIW12, RY13, FKV13, BOSY13]. We survey related work in more detail in Sections 1.1 and 1.2.

1.1 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 with range $[0,1]$, the optimal ϵ-approximating junta has size $\tilde{O}(1/\epsilon^2)$. This is an exponential improvement over the bounds in [FKV13, BOSY13] 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 $[0,1]$ 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 generalization 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.

1.1.1 Structural results

Our main structural result is an approximation of submodular functions by juntas.

Theorem 1.1. *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 $\|f - g\|_2 \leq \epsilon$.*

We also show that this result extends to arbitrary product distributions, with a dependence on the bias of the distribution (see Appendix A). In the special case of submodular functions that take values in $\{0, 1, \ldots, k\}$, our result can be simplified to give a junta of size $O(k \log(k/\epsilon))$ (ϵ being the disagreement probability). This is an exponential improvement over bounds in both [FKV13] and [BOSY13] (see Corollary 3.8 for a formal statement).

Proof technique. Our proof is based on a new procedure that selects variables to be included in the approximating junta for a submodular function f. We view the hypercube $\{0,1\}^n$ 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 of a "*boosting lemma*[1]" on down-monotone events of Goemans and Vondrák [GV06]. 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 the 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 the function being Lipschitz in all the excluded variables which allows us to apply *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.* [BLM00] and are also a crucial component of some of the prior works in this area [BH12, GHRU11, FKV13].

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 a multiplicative (α, ϵ)-approximation to f over a distribution D, if $\Pr_D[f(x) \leq g(x) \leq \alpha f(x)] \geq 1 - \epsilon$. In the PMAC learning model, introduced by Balcan and Harvey [BH12] 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:

Theorem 1.2. *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. 4.1 for definitions). We prove the result via the following generalization of the well-known Friedgut's theorem for boolean functions.

Theorem 1.3. *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 $\|f - g\|_2 \leq \epsilon$. For a submodular, XOS or self-bounding $f : \{0,1\}^n \to [0,1]$, $\mathsf{Infl}^2(f) \leq \mathsf{Infl}^1(f) = O(1)$, 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, the 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 [OS07]. 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 [OS07] generalized the Friedgut's theorem to bounded discretized real-valued functions. They prove a bound of $2^{O(\mathsf{Infl}^2(f)/\epsilon^2)} \cdot \gamma^{-O(1)}$, 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.

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 Fourier degree $O(1/\epsilon^2)$. For the special case of submodular

[1] The terminology comes from [GV06] and has no connection with the notion of boosting in machine learning.

Class of functions	junta size lower bound	junta size upper bound
linear	$\Omega(1/\epsilon^2)$ [Folkl., see Lem. 5.1]	$O(1/\epsilon^2)$ [Folkl.]
coverage	as above	$O(1/\epsilon^2)$ [FK14]
submodular	as above	$O(1/\epsilon^2 \cdot \log(1/\epsilon))$ [Thm. 1.1]
XOS and self-bounding	$2^{\Omega(1/\epsilon)}$ [Thm. 5.2]	$2^{O(1/\epsilon^2)}$ [Thm. 1.3]
constant total ℓ_1-influence	$2^{\Omega(1/\epsilon)}$ [Fri98]	$2^{O(1/\epsilon^2)}$ [Thm. 1.3]
constant total ℓ_2^2-influence	$\Omega(n)$ [OS07]	n

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

functions this was proved by Cheraghchi *et al.* also using Fourier analysis, namely, by bounding the noise stability of submodular functions [CKKL12]. Our more general proof is 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. 5.2). 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.

1.1.2 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 1.4. *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 to 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.*

We remark that this algorithm works even for non-monotone submodular functions and does not in fact rely on our multiplicative-approximation junta result (Theorem 1.2, which works only for monotone submodular functions). Instead, we boostrap the ℓ_2-approximation result (Theorem 1.1) as follows. Theorem 1.1 guarantees an ℓ_2-approximating junta of size $\tilde{O}(1/\epsilon^2)$. The main challenge here is that the criterion for including variables used in the proof of Theorem 1.1 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.

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.) We give the details of these results in Section 6.

For XOS functions, we give a PAC learning algorithm with ℓ_2 error using the junta and low Fourier degree approximation for self-bounding functions (Theorem 1.3).

Theorem 1.5. *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 $\|f - h\|_2 \le \epsilon$. Further, $\mathcal{A}$ runs in time $2^{O(1/\epsilon^4)}\text{poly}(n)$ 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 [Ser04]). Given the influential variables we can exploit the low Fourier degree approximation to find a hypothesis using ℓ_2 regression over the low degree parities (as done in [FKV13]).

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 1.5 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 (see Cor. 6.14 for details). 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.

Organization. We present a detailed discussion of the classes of functions that we consider and technical preliminaries in Section 2. The proof of our main structural result (Thm. 1.1) is presented in Section 3.1. Its extension to multiplicative approximation of monotone submodular functions (Thm. 1.2) is given in Section 3.2. An extension to the case of general product distributions is presented in Appendix A. In Section 4 we give the proof of real-valued generalization of Friedgut's theorem (Thm. 1.3). Section 5 gives examples of functions that prove tightness of our bounds for submodular and XOS functions. The details of our algorithmic applications to PAC and PMAC learning are in Section 6. We state several implications of our structural results to agnostic learning and testing in Section 7.

1.2 Related Work

Reconstruction of submodular functions up to some multiplicative factor (on every point) from value queries was first considered by Goemans *et al.* [GHIM09]. 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 [BDF$^+$12] which studies small-size approximate representations of valuation functions (referred to as *sketches*). Theorem 1.2 shows that allowing an ϵ error probability (over the uniform distribution) makes it possible to get a multiplicative $(1+\gamma)$-approximation using a poly$(1/\gamma, \log(1/\epsilon))$-sized sketch. This sketch can be found in polynomial time using value queries (see Section 3.2).

Balcan and Harvey initiated the study of learning submodular functions from random examples coming from an unknown distribution and introduced the PMAC learning model described above [BH12]. They give an O$(\sqrt{n})$-factor PMAC learning algorithm and show an information-theoretic $\Omega(\sqrt[3]{n})$-factor impossibility result for submodular functions. Subsequently, Balcan *et al.* gave a distribution-independent PMAC learning algorithm for XOS functions that achieves an $\tilde{O}(\sqrt{n})$-approximation and showed that this is essentially optimal [BCIW12]. They also give a PMAC learning algorithm in which the number of clauses defining the target XOS function determines the running time and the 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 [BH12] 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 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 [BH12]. 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 1.2, 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 [GHRU11]. 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 Fourier degree $O(1/\epsilon^2)$ [CKKL12]. 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 [GHRU11] can be computed by a low-rank binary decision tree [FKV13]. 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 [FKV13] 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*) [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 apply Mansour's algorithm for learning DNF [Man95] to 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 [FKV13] 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 [BOSY13].

It is interesting to remark that several largely unrelated methods point to approximating junta being of exponential size, namely, pruned decision trees in [FKV13]; Friedgut's theorem based analysis in this work; two Sunflower lemma-style arguments in [BOSY13]. However, unexpectedly (at least for the authors), a polynomial-size junta suffices.

Previously, approximations by juntas of size polynomial in $1/\epsilon$ were only known in some simple special cases of submodular functions. Boolean submodular functions are disjunctions and hence, over the uniform distribution, can be approximated by an $O(\log(1/\epsilon))$-junta. It can be easily seen that linear functions are approximable by $O(1/\epsilon^2)$-juntas. Coverage functions which are non-negative linear combinations of monotone disjunctions have been recently shown to be approximable by $O(1/\epsilon^2)$-juntas [FK14]. More generally, for Boolean functions the results in [DS09] imply that linear threshold functions with constant total influence can be ϵ-approximated by a junta of size polynomial in $1/\epsilon$. In both [DS09] and [FK14] the techniques are unrelated to ours.

2 Preliminaries

2.1 Classes of valuation functions

Let us describe several classes of functions on the discrete cube, which can be also equivalently viewed as set functions. The functions in these classes share some form of the property of "forbidden complementarities" — e.g., $f(\{a, b\})$ cannot be more than $f(\{a\})+f(\{b\})$. These functions could be monotone or non-monotone; we call a function monotone if $f(S) \leq f(T)$ whenever $S \subset T$.

Linear functions. Linear (or additive) functions are functions in the form $f(S) = \sum_{i \in S} a_i$. This is the smallest class in the hierarchy that we consider here.

Submodular functions. Submodular functions are defined by the condition $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$ for all A, B. A monotone submodular function can be viewed as a valuation on sets with the property of *diminishing returns*: the marginal value of an element, $f_S(i) = f(S \cup \{i\}) - f(S)$, cannot increase if we enlarge the set S. Non-monotone submodular functions play a role in combinatorial optimization, primarily as generalizations of the *cut function* in a graph, $c(S) = |E(S, \bar{S})|$, which is known to be submodular. Another important subclass of monotone submodular functions is the class of *rank functions of matroids*: $r(S) = \max\{|I| : I \in \mathcal{I}, I \subseteq S\}$, where $\mathcal{I}$ is the family of independent sets in a matroid. In fact, it is known that a function of this type is submodular if and only if $\mathcal{I}$ forms a matroid.

Fractionally subadditive functions (XOS). A set function f is fractionally subadditive if $f(A) \leq \sum \beta_i f(B_i)$ whenever $\beta_i \geq 0$ and $\sum_{i:a \in B_i} \beta_i \geq 1$ $\forall a \in A$.

This class is broader than that of (nonnegative) monotone submodular functions (but does not contain non-monotone functions, since fractionally subadditive functions are monotone by definition). For fractionally subadditive functions such that $f(\emptyset) = 0$, there is an equivalent definition known as "XOS" or maximum of non-negative linear functions [Fei06]: f is XOS iff $f(S) = \max_{i \in [m]} \sum_{j \in S} w_{ij}$, where m any positive integer and w_{ij}'s are arbitrary non-negative real-valued weights (note that for every i, $g_i(S) = \sum_{j \in S} w_{ij}$ is a non-negative linear function).

It is instructive to consider again the example of rank functions: $r(S) = \max\{|I| : I \in \mathcal{I}, I \subseteq S\}$. As we mentioned, $r(S)$ is submodular exactly when $\mathcal{I}$ forms a matroid. In contrast, $r(S)$ is XOS for *any down-*

closed set system $\mathcal{I}$ (satisfying $A \subset B \in \mathcal{I} \Rightarrow A \in \mathcal{I}$; this follows from an equivalent formulation of a rank function for down-closed set systems, $r(S) = \max\{|S \cap I| : I \in \mathcal{I}\}$). In this sense, XOS is a significantly broader class than submodular functions. Another manifestation of this fact is that optimization problems like $\max\{f(S) : |S| \leq k\}$ admit constant-factor approximation algorithms using polynomially many *value queries* to f when f is submodular, but no such algorithms exist for XOS functions.

Subadditive functions. Subadditive functions are defined by the condition $f(A \cup B) \leq f(A) + f(B)$ for all A, B. Subadditive functions are more general than submodular and fractionally subadditive functions. In fact, subadditive functions are in some sense much less structured than fractionally subadditive functions. It is easy to verify that every function $f : 2^N \to \{1, 2\}$ is subadditive. While submodular and fractionally subadditive functions satisfy "dimension-free" concentration bounds, this is not true for subadditive functions (see [Von10] for more details).

Self-bounding functions. Self-bounding functions were defined by Boucheron, Lugosi and Massart [BLM00] and further generalized by McDiarmid and Reed [MR06] 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, i.e. the case where $X = \{0, 1\}$. We identify functions on $\{0, 1\}^n$ with set functions on $N = [n]$ in a natural way. By $\mathbf{0}$ and $\mathbf{1}$, we denote the all-zeroes and all-ones vectors in $\{0, 1\}^n$ respectively (corresponding to $\emptyset$ and N sets).

Definition 2.1. *For a function $f : \{0, 1\}^n \to \mathbb{R}$ and any $x \in \{0, 1\}^n$, let $\min_{x_i} f(x) = \min\{f(x), f(x \oplus e_i)\}$. Then f is (a, b)-self-bounding, if for all $x \in \{0, 1\}^n$ and $i \in [n]$,*

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

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

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 non-monotone submodular functions, it is sufficient to consider the slightly more general 2-self-bounding functions — see [Von10]. The 1-Lipschitz condition will not play a role in this paper, as we normalize functions to have values in the $[0, 1]$ range.

Self-bounding functions satisfy *dimension-free concentration bounds*, based on the entropy method of Boucheron, Lugosi and Massart [BLM00]. Currently this is the most general class of functions known to satisfy such concentration bounds. The entropy method for self-bounding functions is general enough to rederive bounds such as Talagrand's concentration inequality. An 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. Another well-studied class of self-bounding functions arises from Rademacher averages which are widely used to measure the complexity of model classes in statistical learning theory [Kol01, BM02]. See [BLB03] for a more detailed discussion and additional examples.

The definition of self-bounding functions is more symmetric than that of submodular functions: note that the definition does not change if we swap the meaning of 0 and 1 for any coordinate. This is a natural property in the setting of machine learning; the learnability of functions on $\{0, 1\}^n$ should not depend on switching the meaning of 0 and 1 for any particular coordinate.

2.2 Norms and discrete derivatives

The ℓ_1 and ℓ_2-norms of $f : \{0, 1\}^n \to \mathbb{R}$ 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 2.2 (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 $x_{i \leftarrow 1, j \leftarrow 1}$.

Absolute error vs. error relative to norm: In our results, we typically assume that the values of $f(x)$ are in a bounded interval $[0,1]$, and our goal is to learn f with an additive error of ϵ. Some prior work considered an error relative to the norm of f, for example at most $\epsilon\|f\|_1$ [CKKL12]. In fact, it is known that for a non-negative submodular or XOS function f, $\|f\|_1 = \mathbf{E}[f] \geq \frac{1}{4}\|f\|_\infty$ [Fei06, FMV07a] and hence this does not make much difference. If we scale $f(x)$ by $\frac{1}{4\|f\|_1}$, we obtain a function with values in $[0,1]$ and learning the original function within an additive error of $\epsilon\|f\|_1$ is equivalent to learning the scaled function within an error of $\epsilon/4$.

3 Junta Approximations of Submodular Functions

First we turn to the class of submodular functions and their approximations by functions of a small number of variables.

3.1 Additive Approximation For Submodular Functions

Here we prove Theorem 1.1, 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 in this paper all logarithms are base 2.

Lemma 3.1. *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 3.1 reduces the number of variables to $O(\log n)$ rather than a constant. However, we show that this is enough to prove Theorem 1.1, effectively by repeating this argument. In fact, it was previously shown [FKV13] that submodular functions can be ϵ-approximated by functions of $2^{O(1/\epsilon^2)}$ variables. One application of Lemma 3.1 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 1.1. Nevertheless, we do not need to rely on this previous result, and we can derive Theorem 1.1 directly from Lemma 3.1 as follows.

Proof of Theorem 1.1. Let $f : \{0,1\}^n \to [0,1]$ be a submodular function. We shall prove a bound of $|J| \leq \frac{4000}{\epsilon^2} \log \frac{1}{\epsilon}$ for the size of the approximating junta.

Observe that this bound holds trivially for $\epsilon \leq n^{-1/2}$, because then we are allowed to choose $J = [n]$. For contradiction, suppose that there is $\epsilon \in (n^{-1/2}, 1/2)$ for which the statement of Theorem 1.1 does not hold. Let $\mathcal{E} \subseteq (n^{-1/2}, 1/2)$ be the set of all ϵ for which the statement does not hold, and pick an $\epsilon \in \mathcal{E}$ such that $\epsilon < 2 \inf \mathcal{E}$. Then, the statement still holds for $\epsilon_2 = \epsilon^2 < \frac{1}{2}\epsilon$.

By the statement of Theorem 1.1 for ϵ_2, there is a subset of variables J of size $|J| \leq \frac{4000}{\epsilon_2^2} \log \frac{1}{\epsilon_2} = \frac{4000}{\epsilon^4} \log \frac{1}{\epsilon^2} \leq \frac{2^{13}}{\epsilon^5}$ and a submodular function g depending only on J, such that $\|f - g\|_2 \leq \epsilon_2 \leq \frac{1}{2}\epsilon$. Now let us apply Lemma 3.1 to g with parameter ϵ. Thus, there exists a submodular function h such that $\|g - h\|_2 \leq \frac{1}{2}\epsilon$, and h depends only on a subset of variables $J' \subseteq J$, $|J'| \leq \frac{128}{\epsilon^2} \log \frac{16|J|}{\epsilon}$. We have $|J| \leq \frac{2^{13}}{\epsilon^5}$, and therefore $|J'| \leq \frac{128}{\epsilon^2} \log \frac{2^{17}}{\epsilon^6} \leq \frac{128}{\epsilon^2} \log \frac{1}{\epsilon^{23}}$ (using $\epsilon \leq \frac{1}{2}$). We conclude that $|J'| \leq \frac{128 \cdot 23}{\epsilon^2} \log \frac{1}{\epsilon} \leq \frac{4000}{\epsilon^2} \log \frac{1}{\epsilon}$ as required in Theorem 1.1. By the triangle inequality, we have $\|f - h\|_2 \leq \|f - g\|_2 + \|g - h\|_2 \leq \frac{1}{2}\epsilon + \frac{1}{2}\epsilon = \epsilon$. However, this would mean that the statement of Theorem 1.1 holds for ϵ as well, which is a contradiction. □

In the rest of this section, our goal is to prove Lemma 3.1.

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 $O(\sqrt{\mathbf{E}[f]})$ [BLM00, Von10]. This fact was also used in previous work on learning of submodular functions [BH12, GHRU11, FKV13]. 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 (the first part is stated as Corollary 2 in [BLB03], Section 2.2; the second part follows easily from the same proof).

Lemma 3.2. *For any self-bounding function $f : \{0,1\}^n \to \mathbb{R}_+$ under a product distribution,*

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

For any a-self-bounding function $f : \{0,1\}^n \to \mathbb{R}_+$ under a product distribution,

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

We use the fact that 1-Lipschitz monotone submodular functions are self-bounding, and 1-Lipschitz nonmonotone submodular functions are 2-self-bounding (see [Von10]). By scaling, we obtain the following for α-Lipschitz submodular functions (see also [FKV13]).

Corollary 3.3. *For any α-Lipschitz monotone submodular function $f : \{0,1\}^n \to \mathbb{R}_+$ under a product distribution,*

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

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

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

Boosting lemma for down-monotone events. The following was proved as Lemma 3 in [GV06].

Lemma 3.4. *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 $p \in (0,1)$.

The proof of Lemma 3.1 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 3.5. *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 3.6. *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 3.5 by a probability bound exponentially small in $1/\delta$. The tool that we use here is the "boosting lemma" (Lemma 3.4) 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 3.7. *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 3.4, we have $\sigma_\delta \leq 1/2$. Therefore, by Lemma 3.4, $\sigma_p = (1-p)^{\phi(p)}$ 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 3.4 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 3.1. 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 3.6 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, $f_{x_{J'}}(y) = f(x_{J'}, y)$. 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:

$$h(x) = \mathbf{E}[f_{x_{J'}}] = \mathbf{E}_{y \in \{0,1\}^{\bar{J'}}}[f(x_{J'}, y)].$$

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

$$\begin{aligned} \|f-h\|_2^2 &= \mathbf{E}_{x \in \{0,1\}^J}[(f(x) - h(x))^2] \\ &= \mathbf{E}_{x_{J'} \in \{0,1\}^{J'}} \mathbf{E}_{y \in \{0,1\}^{\bar{J'}}}[(f(x_{J'}, y) - h(x_{J'}, y))^2] \\ &= \mathbf{E}_{x_{J'} \in \{0,1\}^{J'}} \mathbf{E}_{y \in \{0,1\}^{\bar{J'}}}[(f_{x_{J'}}(y) - \mathbf{E}[f_{x_{J'}}])^2] \\ &= \mathbf{E}_{x_{J'} \in \{0,1\}^{J'}}[\mathbf{Var}[f_{x_{J'}}]]. \end{aligned}$$

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
   - $\partial_i f(x_{J'}) \le \alpha$, and
   - $\partial_i f(x_{J'} + \mathbf{1}_{J \setminus J'}) \ge -\alpha$.

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 Corollary 3.3,

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

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 3.7: 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 3.7, and the distribution of $x_{J'} + \mathbf{1}_{J \setminus J'}$ is the same as $\mathbf{1}_{J \setminus J'(1/2)}$. By Lemma 3.7, 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:

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

Hence, we conclude that $\|f-h\|_2 \le \frac{1}{2}\epsilon$ 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 [RY13] and [BOSY13]. By choosing $\alpha = \frac{1}{k+1}$ and $\delta = 1/(2 \log \frac{2|J|}{\epsilon})$ in the proof above we will obtain that an 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 $2 \cdot 2^{1/(2\delta)} \cdot |J| \le \epsilon$ 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 3.8. *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 $\Pr_{\mathcal{U}}[f \neq g] \leq \epsilon$.*

3.2 Multiplicative Approximation for Monotone Submodular Functions

In this section we show how our approximation theorem can be extended to multiplicative approximation with high probability as required by the PMAC model, introduced by Balcan and Harvey [BH12]. We prove that for any $\gamma > 1, \epsilon > 0$, a multiplicative (γ, ϵ)-approximation for monotone submodular functions over the uniform distribution can be achieved by a function h of a subset of variables whose cardinality depends only on γ and ϵ. More precisely, we prove the following.

Theorem 3.9 (restatement of Theorem 1.2)**.** *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 $J \subseteq [n], |J| \leq \frac{2^{12}}{\gamma^2} \log \frac{16}{\gamma\epsilon} \log \frac{4}{\epsilon}$ such that h is a multiplicative $(1+\gamma, \epsilon)$-approximation of f over the uniform distribution. The function h can be found with high probability using* $\mathrm{poly}(n)$ *value queries to f.*

Observe that (for monotone submodular functions and ignoring the additional logarithm) this is stronger than Theorem 1.1: For any function with range $[0,1]$, a multiplicative $(1+\epsilon, \epsilon)$-approximation implies an additive error bounded by ϵ, except for probability measure of ϵ, which means the ℓ_1 error is bounded by 2ϵ.

The proof of Theorem 1.2 is algorithmic and uses several ideas from the proof of Theorem 1.1. Again, we rely on the boosting lemma and concentration of submodular functions. However, the requirement of a multiplicative approximation to the target function leads to additional complications that we have been able to resolve only in the case of monotone submodular functions. We are not sure whether the theorem holds for non-monotone submodular functions, which we leave as an open question.

As in the case of ℓ_2-error, to prove Theorem 1.2 it is sufficient to prove the following statement.

Lemma 3.10. *For every monotone submodular function $f : \{0,1\}^J \to \mathbb{R}_+$ 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 $J' \subseteq J, |J'| \leq \frac{2^9}{\gamma^2} \log \frac{4}{\epsilon} \log \frac{2|J|}{\epsilon}$ such that h is a multiplicative $(1+\gamma, \epsilon)$-approximation of f over the uniform distribution.*

We find the desired set of significant variables by the following procedure, a modification of Algorithm 3.5.

Algorithm 3.11. *Given $f : \{0,1\}^J \to [0,1]$, produce $J' \subseteq J$ as follows (for parameters $\beta, \delta > 0$):*

- *Set $S := \emptyset$. We use $S(\delta)$ to denote a random subset of S where each element appears independently with probability δ.*
- *As long as there is $i \notin S$ such that*

$$\Pr_{T \sim S(\delta)}[\partial_i f(\mathbf{1}_T) > \beta f(\mathbf{1}_{T \cup \bar{S}})] > \frac{1}{2}$$

  include i in S and repeat.

- *Return $J' := S$.*

The intuition here is that variables get included in S based on their contribution relative to $f(\mathbf{1}_{T \cup \bar{S}}) = f(\mathbf{1}_{S(\delta) \cup \bar{S}})$. Note that this is the top of the subcube defined by fixing the coordinates on S to be equal to $x_S = \mathbf{1}_T$. This is important for obtaining a decomposition such that in each such subcube, the function is sufficiently smooth relative to its own expectation and hence approximated by a constant within a small multiplicative factor. On the other hand, we can bound the number of variables that can be included in S as follows.

Lemma 3.12. *The cardinality of the set J' returned by Algorithm 3.11 is at most $2/(\beta\delta)$.*

Proof. Consider the ordering of elements as they were selected by the algorithm, and assume w.l.o.g. that the ordering is $\{1, 2, 3, \ldots, |J'|\}$. Whenever an element i is included, it is because $\Pr_{T \sim S(\delta)}[\partial_i f(\mathbf{1}_T) > \beta f(\mathbf{1}_{T \cup \bar{S}})] > \frac{1}{2}$. Here, S is the set of elements selected before i, that is $S = [i-1]$ in our ordering.

Thus we can write $T = S(\delta) = R \cap [i-1]$, where $R = J'(\delta)$. The condition above can be written as $\Pr_{R \sim J'(\delta)}[\partial_i f(\mathbf{1}_{R \cap [i-1]}) > \beta f(\mathbf{1}_{R \cup \overline{[i-1]}})] > \frac{1}{2}$. For each $R \subseteq J'$, let us define R^+ as

$$R^+ = \{i \in R : \partial_i f(\mathbf{1}_{R \cap [i-1]}) > \beta f(\mathbf{1}_{R \cup \overline{[i-1]}})\}.$$

Observe that by a telescoping sum,

$$f(\mathbf{1}_R) = f(\mathbf{0}) + \sum_{i \in R} \partial_i f(\mathbf{1}_{R \cap [i-1]}) > \beta \sum_{i \in R^+} \partial_i f(\mathbf{1}_{R \cup \overline{[i-1]}}) \geq |R^+| \cdot \beta f(\mathbf{1}_R)$$

and hence $|R^+| < 1/\beta$ for every R.

Consider the expectation $\mathbf{E}_{R \sim J'(\delta)}[|R^+|]$. As we argued above, every time we include i in J', we have the property that $\Pr_{R \sim J'(\delta)}[f(\mathbf{1}_{R \cap [i-1]}) > \beta f(\mathbf{1}_{R \cup \overline{[i-1]}})] > \frac{1}{2}$. Since i appears in R with probability δ, independently of the condition $\partial_i f(\mathbf{1}_{R \cap [i-1]}) > \beta f(\mathbf{1}_{R \cup \overline{[i-1]}})$, this means that each element $i \in J'$ appears in R^+ with probability at least $\delta/2$. We conclude that $\mathbf{E}_{R \sim J'(\delta)}[|R^+|] \geq |J'|\delta/2$. On the other hand, $|R^+| < 1/\beta$ for all R. This implies that $|J'| < 2/(\beta\delta)$. □

Recall that so far, we were working with subsets of J' sampled with a (small) probability δ. The next step is to prove that for a *uniformly* random assignment $x_{J'} \in \{0,1\}^{J'}$, the function $f_{x_{J'}}(y) = f(x_{J'}, y)$ for $y \in \{0,1\}^{\bar{J}'}$ has suitable Lipschitz properties for most values of $x_{J'}$. This relies on the boosting lemma, and in this step we require again that f is a *monotone* submodular function. In the following, all expectations are over a uniform distribution on the respective subcube, unless otherwise indicated.

Lemma 3.13. *The set J' returned by Algorithm 3.11 satisfies for every $i \notin J'$,*

$$\Pr_{x_{J'} \in \{0,1\}^{J'}}[\partial_i f(x_{J'}) > \beta f(x_{J'}, \mathbf{1}_{\bar{J}'})] \leq 2^{-1/(2\delta)}.$$

Proof. Denote by $\mathcal{F} \subseteq \{0,1\}^{J'}$ the family of points $x_{J'}$ such that the condition is satisfied, i.e. $\mathcal{F} = \{x_{J'} \in \{0,1\}^{J'} : \partial_i f(x_{J'}) > \beta f(x_{J'}, \mathbf{1}_{\bar{J}'})\}$. This is a down-monotone set: if $y \leq x \in \mathcal{F}$, then $y \in \mathcal{F}$ because $\partial_i f(y) \geq \partial_i f(x)$, and $f(y, \mathbf{1}_{\bar{J}'}) \leq f(x, \mathbf{1}_{\bar{J}'})$ (here we are using both monotonicity and submodularity).

If we define $\sigma_p = \Pr[\mathbf{1}_{J'(p)} \in \mathcal{F}]$, this means that $\sigma_\delta \leq 1/2$. By Lemma 3.4, we have $\Pr[\mathbf{1}_{J'(1/2)} \in \mathcal{F}] = \sigma_{1/2} \leq 2^{-1/(2\delta)}$. As $\mathbf{1}_{J'(1/2)}$ is distributed uniformly in $\{0,1\}^{J'}$, this is exactly the statement of Lemma 3.13. □

Finally, we finish the proof of Lemma 3.10 by using concentration properties of submodular functions. We refer to the following bound from [Von10].

Lemma 3.14. *If $Z = f(X_1, \ldots, X_n)$ where $X_i \in \{0,1\}$ are independently random and f is a nonnegative submodular function with discrete derivatives bounded by $[-1,1]$, then for any $\lambda > 0$,*

- $\Pr[Z \geq (1+\lambda)\mathbf{E}[Z]] \leq e^{-\lambda^2 \mathbf{E}[Z]/(4+5\lambda/3)}$.
- $\Pr[Z \leq (1-\lambda)\mathbf{E}[Z]] \leq e^{-\lambda^2 \mathbf{E}[Z]/4}$.

Proof of Lemma 3.10. Given a monotone submodular function $f : \{0,1\}^J \to \mathbb{R}_+$ and $\gamma, \epsilon \in (0,1)$, we construct $J' \subseteq J$ by running Algorithm 3.11 with parameters $\beta = \frac{1}{108}\gamma^2/\log\frac{4}{\epsilon}$ and $\delta = 1/(2\log\frac{2|J|}{\epsilon})$. By Lemma 3.12, the constructed subset of variables has size $|J'| \leq 2/(\beta\delta) \leq 2^9\gamma^{-2}\log\frac{4}{\epsilon}\log\frac{2|J|}{\epsilon}$.

By Lemma 3.13, we obtain a subset of variables J' such that for every $i \notin J'$,

$$\Pr_{x_{J'} \in \{0,1\}^{J'}}[\partial_i f(x_{J'}) > \beta f(x_{J'}, \mathbf{1}_{\bar{J}'})] \leq \frac{\epsilon}{2|J|}.$$

By the union bound,

$$\Pr_{x_{J'} \in \{0,1\}^{J'}}[\exists i \in J \setminus J'; \partial_i f(x_{J'}) > \beta f(x_{J'}, \mathbf{1}_{\bar{J}'})] \leq \frac{\epsilon}{2}.$$

This means that with probability $1-\epsilon/2$ over the choice of $x_{J'} \in \{0,1\}^{J'}$, the point $x_{J'}$ is *good* in the sense that the function $f_{x_{J'}}(y) = f(x_{J'}, y)$ for $y \in \{0,1\}^{\bar{J}'}$ has discrete derivatives bounded by $\partial_i f(x_{J'}) \leq \beta f_{x_{J'}}(\mathbf{1}_{\bar{J}'})$. Fix any good point $x_{J'}$. By submodularity, the same bound holds for the derivatives evaluated at any point above $x_{J'}$. In addition, f is monotone, hence $\partial_i f_{x_{J'}}(y) \in [0, \beta f_{x_{J'}}(\mathbf{1}_{\bar{J}'})]$ for all $y \in \{0,1\}^{\bar{J}'}$.

Here we use a concentration bound for submodular functions (Lemma 3.14). Consider the function $f_{x_{J'}}$ for a good point $x_{J'}$. We apply the concentration bound to a scaled function $\tilde{f}(y) = f_{x_{J'}}(y)/(\beta f_{x_{J'}}(\mathbf{1}_{\bar{J}'}))$. By the discussion above, $\tilde{f}$ has discrete derivatives in $[0,1]$. By Lemma 3.14, for $\lambda \in [0,1]$,

$$\Pr_{y \in \{0,1\}^{\bar{J}'}}[|\tilde{f}(y) - \mathbf{E}[\tilde{f}]| > \lambda \mathbf{E}[\tilde{f}]] < 2e^{-\lambda^2 \mathbf{E}[\tilde{f}]/6}.$$

We also use a known fact [Fei06] that for any monotone submodular function, $\mathbf{E}[\tilde{f}] \geq \frac{1}{2}\|\tilde{f}\|_\infty = \frac{1}{2}\tilde{f}(\mathbf{1}_{\bar{J}'}) = 1/(2\beta)$. Going back to $f_{x_{J'}}$, we obtain

$$\Pr_{y \in \{0,1\}^{\bar{J}'}}[|f_{x_{J'}}(y) - \mathbf{E}[f_{x_{J'}}]| > \lambda \mathbf{E}[f_{x_{J'}}]] < 2e^{-\lambda^2/(12\beta)}.$$

We set $\lambda = \gamma/3$, and recall that we have $\beta = \frac{1}{108}\gamma^2/\log\frac{4}{\epsilon}$. Therefore

$$\Pr_{y \in \{0,1\}^{\bar{J}'}}[|f_{x_{J'}}(y) - \mathbf{E}[f_{x_{J'}}]| > \frac{1}{3}\gamma \mathbf{E}[f_{x_{J'}}]] < 2e^{-\log\frac{4}{\epsilon}} \leq \frac{\epsilon}{2}$$

for every good point $x_{J'}$. Equivalently,

$$\Pr_{y \in \{0,1\}^{\bar{J}'}}\left[\frac{f_{x_{J'}}(y)}{1+\gamma/3} \leq \mathbf{E}[f_{x_{J'}}] \leq \frac{f_{x_{J'}}(y)}{1-\gamma/3}\right] > 1 - \frac{\epsilon}{2} \tag{3}$$

for every good point $x_{J'}$.

We define our approximation to f as follows:

$$h(x) = \left(1 + \frac{\gamma}{3}\right)\mathbf{E}[f_{x_{J'}}].$$

In other words, we average out the contributions of all variables outside of J', and we adjust by a constant factor of $1 + \frac{\gamma}{3}$, to make sure that $h(x) \geq f(x)$ with the desired probability. Observe that h is a positive linear combination of monotone submodular functions, and hence also a monotone submodular function. Also, h depends only on the variables in J'.

Now, our goal is to estimate the probability that $f(x) \leq h(x) \leq (1+\gamma)f(x)$. In the following, all probabilities and expectations are over uniform distributions. We have

$$\begin{aligned}
& \Pr_{x \in \{0,1\}^J}[f(x) \leq h(x) \leq (1+\gamma)f(x)] \\
= \;& \mathbf{E}_{x_{J'} \in \{0,1\}^{J'}}\left[\Pr_{y \in \{0,1\}^{\bar{J}'}}[f_{x_{J'}}(y) \leq h(x_{J'}) \leq (1+\gamma)f_{x_{J'}}(y)]\right] \\
= \;& \mathbf{E}_{x_{J'} \in \{0,1\}^{J'}}\left[\Pr_{y \in \{0,1\}^{\bar{J}'}}[f_{x_{J'}}(y) \leq (1+\gamma/3)\mathbf{E}[f_{x_{J'}}] \leq (1+\gamma)f_{x_{J'}}(y)]\right] \\
\geq \;& \mathbf{E}_{x_{J'} \in \{0,1\}^{J'}}\left[\Pr_{y \in \{0,1\}^{\bar{J}'}}\left[\frac{f_{x_{J'}}(y)}{1+\gamma/3} \leq \mathbf{E}[f_{x_{J'}}] \leq \frac{f_{x_{J'}}(y)}{1-\gamma/3}\right]\right] \\
\geq \;& \Pr_{x_{J'} \in \{0,1\}^{J'}}[x_{J'} \text{ is good}] \cdot \left(1 - \frac{\epsilon}{2}\right)
\end{aligned}$$

using Eq. (3). As we argued above, a uniformly random point $x_{J'} \in \{0,1\}^{J'}$ is good with probability at least $1 - \epsilon/2$. Hence,

$$\Pr_{x \in \{0,1\}^J}[f(x) \leq h(x) \leq (1+\gamma)f(x)] \geq \left(1 - \frac{\epsilon}{2}\right)^2 \geq 1 - \epsilon$$

which is the definition of multiplicative $(1+\gamma, \epsilon)$-approximation. □

Now we can finish the proof of Theorem 1.2.

Proof of Theorem 1.2. Given a monotone submodular function $f : \{0,1\}^n \to \mathbb{R}_+$, we use Lemma 3.10 repeatedly to reduce the number of variables. To work out the necessary parameters, we proceed backwards: Eventually, we want to obtain a multiplicative $(1+\gamma, \epsilon)$-approximation, using $O(\frac{1}{\gamma^2}\log\frac{1}{\gamma\epsilon}\log\frac{1}{\epsilon})$ variables. Let us define the following sequences: (for a constant c to be determined later)

- $\gamma_i = \gamma/2^i$, $\epsilon_i = \epsilon/2^i$,
- $n_0 = \lfloor \frac{c}{\gamma^2} \log \frac{16}{\gamma\epsilon} \log \frac{4}{\epsilon} \rfloor$,
- $n_{i+1} = \lfloor \frac{\epsilon_i}{2} \cdot 2^{n_i \gamma_i^2 / (2^9 \log \frac{4}{\epsilon_i})} \rfloor$.

The meaning of this sequence is that given a function f_{i+1} of n_{i+1} variables and parameters γ_i, ϵ_i, we can find a function f_i of n_i variables which is a multiplicative $(1+\gamma_i, \epsilon_i)$-approximation of f_{i+1} (using Lemma 3.10, and inverting the relationship between n_i and n_{i+1}). Note that the parameters γ_i, ϵ_i form geometric series adding up to at most γ and ϵ respectively, and consequently f_0 is a multiplicative $(1+\gamma, \epsilon)$-approximation of f_k for any $k > 0$.

By induction, we prove the following for every $i \geq 0$:

$$n_i \geq \lfloor c \frac{1}{\gamma_i^2} \log \frac{16}{\gamma_i \epsilon_i} \log \frac{4}{\epsilon_i} \rfloor. \tag{4}$$

The base case holds by definition. So assume that (4) holds for n_i. For n_{i+1}, we obtain

$$\begin{aligned} n_{i+1} &= \lfloor \frac{\epsilon_i}{2} \cdot 2^{n_i \gamma_i^2 / (2^9 \log \frac{4}{\epsilon_i})} \rfloor \\ &\geq \lfloor \frac{\epsilon_i}{2} \cdot 2^{c(\log \frac{16}{\gamma_i \epsilon_i})/2^{10}} \rfloor \\ &= \left\lfloor \frac{\epsilon_i}{2} \cdot \left(\frac{16}{\gamma_i \epsilon_i} \right)^{c/2^{10}} \right\rfloor . \end{aligned}$$

We pick $c = 2^{14}$, and use $\gamma_i = 2\gamma_{i+1}, \epsilon_i = 2\epsilon_{i+1}$, which yields

$$\begin{aligned} n_{i+1} &\geq \left\lfloor \epsilon_{i+1} \cdot \left(\frac{4}{\gamma_{i+1} \epsilon_{i+1}} \right)^{16} \right\rfloor \\ &= \lfloor 2^{32} \gamma_{i+1}^{-16} \epsilon_{i+1}^{-15} \rfloor \\ &\geq \left\lfloor \frac{2^{14}}{\gamma_{i+1}^2} \log \frac{16}{\gamma_{i+1} \epsilon_{i+1}} \log \frac{4}{\epsilon_{i+1}} \right\rfloor . \end{aligned}$$

This proves Eq. (4). Note that in particular, since $\gamma_i = \gamma/2^i, \epsilon_i = \epsilon/2^i$, this proves that n_i grows at least as a geometric sequence, and will reach $n_t \geq n$ in $t = O(\log n)$ steps (in fact much faster, but we are not concerned with the exact number of iterations). Therefore, we can take $f_t = f$ to be our original function and work our way backwards, to obtain a multiplicative $(1+\gamma, \epsilon)$-approximation f_0 which depends on at most $n_0 = \lfloor \frac{2^{14}}{\gamma^2} \log \frac{16}{\gamma\epsilon} \log \frac{4}{\epsilon} \rfloor$ variables.

We remark that the proof is constructive and we have in fact constructed the multiplicative junta approximation by a randomized polynomial-time algorithm (with value query access to f) that succeeds with high probability. □

4 Approximation of Low-Influence Functions by Juntas

Here we show how structural results for submodular (weaker than the one in Section 3.1), XOS and self-bounding functions can be proved in a unified manner using the notion of total influence.

4.1 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 *Fourier degree* of f is the largest $|S|$ such that $\hat{f}(S) \neq 0$. 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. [BOL85, KKL88]).

Definition 4.1 (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 $2\cdot\mathsf{Infl}^1(f)$ 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 $\mathsf{Infl}^2(f) = \sum_S |S|\hat{f}^2(S)$.

4.2 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 4.2. *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 an XOS function $f : \{0,1\}^n \to [0,1]$, $\mathsf{Infl}^1(f) \leq 1$. For a submodular $f : \{0,1\}^n \to [0,1]$, $\mathsf{Infl}^1(f) \leq 2$.*

Proof. We have

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

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 [Von10]). □

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 4.3. *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 [CKKL12]). Next, we show a stronger statement, that these functions are ϵ-approximated by $2^{O(1/\epsilon^2)}$-juntas of Fourier degree $O(1/\epsilon^2)$.

4.3 Friedgut's Theorem for Real-Valued Functions

As we have shown in Lemma 4.2, self-bounding functions have low total ℓ_1-influence. A celebrated result of Friedgut [Fri98] 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. [DFKO06]). However, a Friedgut-style result is not true for real-valued functions when ℓ_2^2-influences are used, as observed by O'Donnell and Servedio [OS07] (see also Sec. 5.3). This issue also arises in the problem of learning real-valued monotone decision trees [OS07]. 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 parameters 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 holds for real-valued functions if the total ℓ_κ^κ-influence (for some constant $\kappa \in [1,2)$) is small in addition to total ℓ_2^2-influence. 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 4.4. *Let $f : \{0,1\}^n \to \mathbb{R}$ be any function, $\epsilon \in (0,1)$ and $\kappa \in (1,2)$. For d such that $\sum_{|S|>d} \hat{f}(S)^2 \leq \epsilon/2$, let*

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

$$\alpha = \left((\kappa-1)^{d-1} \cdot \epsilon/(2 \cdot \mathsf{Infl}^\kappa(f))\right)^{\kappa/(2-\kappa)} \ .$$

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 1.3 from this statement we use it with ϵ^2 error and let $d = 2 \cdot \mathsf{Infl}^2(f)/\epsilon^2$ which, by Lemma 4.3, gives the desired bound on $\sum_{|S|>d} \hat{f}(S)^2$. 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 at most

$$|I| \leq \mathsf{Infl}^\kappa(f)/\alpha = 2^{O(\mathsf{Infl}^2(f)/\epsilon^2)} \cdot \epsilon^{2\kappa/(2-\kappa)} \cdot (\mathsf{Infl}^\kappa(f))^{2/(2-\kappa)}. \tag{5}$$

Also note that Theorem 4.4 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 (5) 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 1.3 (first part). The second part of Theorem 1.3 now follows from Lemma 4.2.

Our proof of Theorem 4.4 is a simple modification of the proof of Friedgut's theorem from [DF05]. We will need the notion of a noise operator.

Definition 4.5 (The noise operator). *For $\alpha \in [0,1], x \in \{0,1\}^n$, we define a distribution $N_\alpha(x)$ over $y \in \{0,1\}^n$ by letting $y_i = x_i$ with probability $1-\alpha$ and $y_i = 1 - x_i$ with probability α, independently for each i. For $\rho \in [-1,1]$, the noise operator T_ρ acts on functions $f : \{0,1\}^n \to \mathbb{R}$, and is defined by*

$$(T_\rho f)(x) = \mathbf{E}_{y \sim N_{1/2-\rho/2}(x)}[f(y)].$$

In the Fourier basis the noise operator satisfies: $\widehat{(T_\rho f)}(S) = \rho^{|S|}\hat{f}(S)$, for every $S \subseteq [n]$.

Following Friedgut's proof, we will require a bound on $\|T_\rho f\|_2$ in terms of $\|f\|_\kappa$. This lemma is a special case of the Hypercontractive inequality of Bonami and Beckner [Bon70, Bec75].

Lemma 4.6. *For any $f : \{0,1\}^n \to \mathbb{R}$, and any $\kappa \in [1,2]$, $\|T_{\sqrt{\kappa-1}} f\|_2 \leq \|f\|_\kappa$.*

The proof of Theorem 4.4 relies on two lemmas. The first one is Lemma 4.3, stated above. The second and key lemma is the following bound on the sum of squares of all low-degree Fourier coefficients that include a variable of low influence.

Lemma 4.7. *Let $f : \{0,1\}^n \to \mathbb{R}$, $\kappa \in (1,2)$, $\alpha > 0$ and d be an integer ≥ 1. Let $I = \{i \in [n] \mid \mathsf{Infl}_i^\kappa(f) \geq \alpha\}$. Then*

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

Proof. We first observe that by the properties of the Fourier transform of $\partial_i f$ (see Sec. 4.1) and the noise operator T_ρ, we have

$$\left\| T_\rho \frac{\partial_i f}{2} \right\|_2^2 = \sum_{S\subseteq[n], S\ni i} (\rho^2)^{|S|-1} \hat{f}(S)^2. \tag{6}$$

Next we bound the sum in terms of norms of $T_{\sqrt{\kappa-1}}$ applied to $\partial_i f$'s.

$$\begin{aligned}\sum_{S\not\subseteq I, |S|\le d} \hat{f}(S)^2 &\le \sum_{S\subseteq[n], |S|\le d} |S\cap \bar{I}| \hat{f}(S)^2 \le (\kappa-1)^{1-d} \sum_{S\subseteq[n], |S|\le d} |S\cap\bar{I}| (\kappa-1)^{|S|-1} \hat{f}(S)^2 \\ &= (\kappa-1)^{1-d} \sum_{i\in\bar{I}} \sum_{S\subseteq[n], S\ni i} (\kappa-1)^{|S|-1} \hat{f}(S)^2 = (\kappa-1)^{1-d} \sum_{i\in\bar{I}} \left\| T_{\sqrt{\kappa-1}} \frac{\partial_i f}{2} \right\|_2^2,\end{aligned}$$

where the last equality follows from eq. (6). Now we can apply Lemma 4.6 to obtain:

$$\begin{aligned}\sum_{i\in\bar{I}} \left\| T_{\sqrt{\kappa-1}} \frac{\partial_i f}{2} \right\|_2^2 \le \sum_{i\in\bar{I}} \left\| \frac{\partial_i f}{2} \right\|_\kappa^2 &= \sum_{i\in\bar{I}} \mathbf{E}\left[\left| \frac{\partial_i f}{2} \right|^\kappa \right]^{\frac{1}{\kappa}\cdot 2} \\ &= \sum_{i\in\bar{I}} (\mathsf{Infl}_i^\kappa(f))^{2/\kappa} \\ &\le \max_{i\in\bar{I}} (\mathsf{Infl}_i^\kappa(f))^{2/\kappa-1} \cdot \sum_{i\in\bar{I}} \mathsf{Infl}_i^\kappa(f) \\ &\le \cdot\alpha^{2/\kappa-1} \cdot \mathsf{Infl}^\kappa(f).\end{aligned}$$

□

We now proceed to obtain Theorem 4.4 by combining Lemmas 4.3 and 4.7.

Proof of Thm. 4.4. Observe that

$$\sum_{S\not\in\mathcal{I}_d} \hat{f}(S)^2 = \sum_{S\subseteq[n], |S|>d} \hat{f}(S)^2 + \sum_{S\not\subseteq I, |S|\le d} \hat{f}(S)^2 \ .$$

For our choice of d, $\sum_{S\subseteq[n], |S|>d} \hat{f}(S)^2 \le \epsilon/2$.

Now, by Lemma 4.7 the second part can be bounded by

$$\sum_{S\not\subseteq I, |S|\le d} \hat{f}(S)^2 \le (\kappa-1)^{1-d} \cdot \alpha^{2/\kappa-1} \cdot \mathsf{Infl}^\kappa(f) = (\kappa-1)^{1-d} \cdot \left((\kappa-1)^{d-1} \cdot \epsilon/(2\mathsf{Infl}^\kappa(f))\right) \cdot \mathsf{Infl}^\kappa(f) = \epsilon/2 \ .$$

□

We now give a slightly simpler version of Thm. 4.4 for functions that have low total ℓ_1-influence, such as self-bounding functions.

Corollary 4.8. *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*

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

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

Proof. We first note that, for every i, $\frac{\partial_i f}{2}$ has range in $[-1,1]$ and therefore for every $\kappa \ge 1$,

$$\mathsf{Infl}_i^\kappa(f) = \mathbf{E}\left[\left| \frac{\partial_i f}{2} \right|^\kappa \right] \le \mathbf{E}\left[\left| \frac{\partial_i f}{2} \right| \right] = \mathsf{Infl}_i^1(f).$$

In particular, $\mathsf{Infl}^2(f) \le \mathsf{Infl}^1(f)$ and $\mathsf{Infl}^{4/3}(f) \le \mathsf{Infl}^1(f)$. We can now apply Thm. 4.4 with $\kappa = 4/3$ to obtain that for $d = 2\cdot\mathsf{Infl}^1(f)/\epsilon^2 \ge 2\cdot\mathsf{Infl}^2(f)/\epsilon^2$, $\alpha = 2^{-4d} \le \left(3^{-d+1}\cdot\epsilon^2/(2\cdot\mathsf{Infl}^{4/3}(f))\right)^2$ and

$$I' = \{i \in [n] \mid \mathsf{Infl}_i^{4/3}(f) \ge \alpha\}$$

we have that

$$\sum_{S \not\subseteq I' \text{ or } |S|>d} (\hat{f}(S))^2 \leq \epsilon^2.$$

Let $p = \sum_{S \subseteq I',\ |S| \leq d} \hat{f}(S)\chi_S$. Then $\|f - p\|_2^2 \leq \epsilon^2$. Now we observe that $\mathsf{Infl}_i^{4/3}(f) \leq \mathsf{Infl}_i^1(f)$ implies that $I' \subseteq I$ and therefore p is a function of Fourier degree d over variables in I. To bound $\|\hat{p}\|_1$ we observe that $|I| \leq \mathsf{Infl}^1(f)/\alpha$ and therefore the total number of non-zero Fourier coefficients of p is at most

$$\sum_{j \leq d} \binom{|I|}{j} \leq |I|^d = (2^{4d} \cdot \mathsf{Infl}^1(f))^d = 2^{O(\mathsf{Infl}^1(f)^2/\epsilon^4)} .$$

To get the desired bound on $\|\hat{p}\|_1$ it now suffices to note that f has range $[-1, 1]$ and therefore for every $S \subseteq [n]$, $|\hat{p}(S)| \leq |\hat{f}(S)| \leq 1$. □

5 Lower Bound Examples

Here we show three simple examples: The first one shows that Theorem 1.1 is almost optimal, in the sense that the dependence on ϵ cannot be better than $1/\epsilon^2$. The second example shows that Corollary 4.8 is essentially optimal even for XOS functions. Finally, the third example shows that Theorem 4.4 requires the use of ℓ_κ^κ-influences for $\kappa < 2$ rather than just ℓ_2^2-influences.

5.1 Lower Bound On Junta Size For Linear Functions

We prove that even for linear functions, an ϵ-approximation (even in ℓ_1-norm) requires at least $1/\epsilon^2$ variables.

Lemma 5.1. *Consider a linear function $f : \{0,1\}^n \to [0,1]$,*

$$f(x) = \frac{1}{a} \sum_{i \in A} x_i$$

where $|A| = a$. Then every function $g : \{0,1\}^n \to \mathbb{R}$ that depends on less than $\frac{a}{2}$ variables has $\|f - g\|_1 = \Omega(\sqrt{1/a})$.

Proof. Suppose that g depends only on a subset of variables B. Denote by f_{x_B} the restriction of f to $\{0,1\}^{\bar{B}}$ after the coordinates on B have been fixed to x_B. Note that f_{x_B} is still a linear function. Hence, the closest function to f depending only on x_B (whether in ℓ_1 or ℓ_2) is $g(x) = \mathbf{E}[f_{x_B}] = \frac{1}{a}\sum_{i \in B} x_i + \frac{1}{2a}|A \setminus B|$.

Let us compute the distance between f and g: After fixing the coordinates on B, f_{x_B} is a linear function of variance

$$\mathbf{Var}[f_{x_B}] = \sum_{i \in A \setminus B} \frac{1}{a^2} \mathbf{Var}[x_i] = |A \setminus B| \cdot \frac{1}{4a^2}.$$

This means that with constant probability, f_{x_B} deviates from its expectation by at least $\sqrt{\mathbf{Var}[f_{x_B}]} = \frac{1}{2a}\sqrt{|A \setminus B|}$. Consequently, $|f(x) - g(x)| > \frac{1}{2a}\sqrt{|A \setminus B|}$ with constant probability and $\|f - g\|_1 = \Omega(\frac{1}{2a}\sqrt{|A \setminus B|})$. If $|A \setminus B| \geq \frac{a}{2}$, then we obtain $\|f - g\|_1 = \Omega(\sqrt{1/a})$. □

5.2 Lower Bound On Junta Size For XOS Functions

Here we prove that Theorem 1.3 is close-to-tight and, in particular, Theorem 1.1 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 [BOL85] with AND replaced by a linear function. The Tribes DNF was also used by Friedgut to prove tightness of his theorem for Boolean functions [Fri98].

Theorem 5.2. *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 $\|f - g\|_1 = \Omega(1/a)$.

Proof. Suppose that g depends on fewer than 2^{a-1} variables. This means that there are fewer than 2^{a-1} parts where g depends on any variable. Let us denote the parts where g does not depend on any variable by $\mathcal{D}$; we have $|\mathcal{D}| > 2^{a-1}$.

We observe the following: For each part, $\Pr[\sum_{i \in A_j} x_i = a] = 2^{-a}$ (all a variables should be equal to 1). Therefore, with probability at least $(1 - 2^{-a})^{2^{a-1}} \simeq e^{-1/2}$ we have $\sum_{i \in A_j} x_i < a$ for all $j \notin \mathcal{D}$. Let us condition on some values of $\{x_i : i \in \bigcup_{j \notin \mathcal{D}} A_j\}$ such that this is the case. In this event, $f(x) = 1$ iff we have $\sum_{i \in A_j} x_i = a$ for at least one of the parts $j \in \mathcal{D}$. This happens with constant probability (since $2^{a-1} < |\mathcal{D}| \leq 2^a$), bounded away from both 0 and 1. Hence, $f(x)$ is either 1 or at most $1 - 1/a$, both with constant nonzero probabilities.

On the other hand, our function g does not depend on the variables in $\bigcup_{j \in \mathcal{D}} A_j$ at all. Therefore, given the variables $\{x_i : i \in \bigcup_{j \notin \mathcal{D}} A_j\}$, $g(x)$ has a fixed value, and with constant probability it differs from $f(x)$ by at least $\frac{1}{2a}$. Overall, this happens with constant probability, and hence $\|f - g\|_1 = \Omega(1/a)$. □

5.3 Lower Bound For Total ℓ_2^2-influence

Here we show that a generalization of Friedgut's theorem to real-valued functions cannot use total ℓ_2^2-influence only. A similar example also appears in [OS07].

Lemma 5.3. *There is an absolute constant $\alpha > 0$ and a function $f : \{-1,1\}^n \to [-1,1]$ for any n, such that $\mathsf{Infl}^2(f) \leq 1$, and for any function g that depends only on $n/2$ variables, $\|f - g\|_1 \geq \alpha$.*

Proof. Let

- $f(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^n x_i$ for $|\sum_{i=1}^n x_i| \leq \sqrt{n}$,
- $f(x) = 1$ for $\sum_{i=1}^n x_i > \sqrt{n}$, and
- $f(x) = -1$ for for $\sum_{i=1}^n x_i < -\sqrt{n}$.

The total ℓ_2^2-influence is easy to estimate:

$$\mathsf{Infl}^2(f) = \sum_{i=1}^n \mathbf{E}[(\frac{1}{2}\partial_i f(x))^2] \leq n \cdot \frac{1}{n} = 1.$$

Now assume that $g : \{-1,1\}^n \to \mathbb{R}$ depends only on a subset of coordinates J, $|J| = n/2$. Condition on any choice of values for x_J such that $|\sum_{i \in J} x_i| < \sqrt{n}$. (This happens with constant probability for random x_J.) The remaining $n/2$ variables satisfy with constant probability $(\mathsf{sign}\ g(x_J))(\sum_{i \notin J} x_i) < -2\sqrt{n}$ (recall that g depends only on x_J). This implies that $(\mathsf{sign}\ g(x)) \cdot \sum_{i=1}^n x_i < -\sqrt{n}$; i.e., $g(x)$ and $f(x)$ have opposite signs and moreover $|\sum_{i=1}^n x_i| > \sqrt{n}$, so $|f(x)| = 1$. Thus with constant probability, $|f(x) - g(x)| \geq 1$. □

6 Applications to PAC Learning

We now show that our approximation of submodular and low-influence functions by juntas can be used to give faster PAC and PMAC learning algorithms for these classes of functions.

6.1 Preliminaries: Models of Learning

We consider two models of learning based on the PAC model [Val84] which assumes that the learner has access to random examples of an unknown function from a known class of functions. In the first model we measure the performance of the learner by ℓ_1-error between the target and the hypothesis, which generalizes the notion of disagreement error used for learning Boolean functions (e.g. [Hau92]).

Definition 6.1 (PAC learning with ℓ_1-error)**.** *Let $\mathcal{F}$ be a class of real-valued functions on $\{0,1\}^n$ and let D be a distribution on $\{0,1\}^n$. An algorithm $\mathcal{A}$ PAC-learns $\mathcal{F}$ on D, if given $\epsilon > 0$, for every target function $f \in \mathcal{F}$, given access to random independent samples from D labeled by f, with probability at least $2/3$, $\mathcal{A}$ returns a hypothesis h such that $\mathbf{E}_{x \sim D}[|f(x) - h(x)|] \leq \epsilon$. $\mathcal{A}$ is said to be* proper *if $h \in \mathcal{F}$. $\mathcal{A}$ is said to be* efficient *if h can be evaluated in polynomial time on any input and the running time of $\mathcal{A}$ is polynomial in n and $1/\epsilon$.*

In some cases we bound the ℓ_2-error of the hypothesis which also upper-bounds its ℓ_1-error. While in general Valiant's model does not make assumptions on the distribution D, here we only consider the *distribution-specific* version of the model in which the distribution is fixed and is uniform over $\{0,1\}^n$.

The second model that we consider is the PMAC model introduced by Balcan and Harvey [BH12] which requires a multiplicative-factor approximation of the target function. A PMAC learner with approximation factor α and error ϵ is an algorithm which outputs a hypothesis h that satisfies $\Pr_{x\sim D}[f(x) \leq h(x) \leq \alpha f(x)] \geq 1-\epsilon$. We say that h multiplicatively (α,ϵ)-approximates f over D in this case.[2]

6.2 Finding Influential Variables

In order to exploit the fact that a submodular function can be approximated by a junta we need to find the variables of the junta. Unfortunately, the criterion for including variables given in Algorithm 3.5 cannot be (efficiently) evaluated using random examples alone. Instead we give a general way to find a larger approximating junta whenever an approximating junta exists. For a real-valued f over $\{0,1\}^n$ and $\epsilon \in [0,1]$ let $s_f(\epsilon)$ denote the smallest s such that there exists an s-junta g for which $\|f-g\|_2 \leq \epsilon$. For a set of indices $I \subseteq [n]$ we say that a function is an I-junta if it depends only on variables in I.

Theorem 6.2. *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 at most $32 \cdot (s_f(\epsilon/2))^2/\epsilon$ such that there exists a submodular J-junta h for $J \subseteq I$ of size $s_f(\epsilon/2)$ satisfying $\|f-h\|_1 \leq \epsilon$. The algorithm runs in time $O(n^2\log(n)\cdot(s_f(\epsilon/2))^4/\epsilon^2)$ and uses $O(\log(n)\cdot(s_f(\epsilon/2))^4/\epsilon^2)$ examples.*

Our algorithm selects all variables that have a large degree-1 or 2 Fourier coefficient. This is the same algorithm as the one used in [FKV13] (with different values of thresholds). However the analysis in [FKV13] relies crucially on the spectral ℓ_1-norm of an ϵ-approximating function g and gives a junta of size $\mathrm{poly}(\|\hat{g}\|_1)$. As can be seen from the lower bound in [FKV13], the spectral ℓ_1-norm of any function that ϵ-approximates certain submodular functions must be exponential in $1/\epsilon$ and therefore this argument is not useful for our purposes. Instead we give a new and more general argument that relies on the fact that total ℓ_1-influence of submodular functions is upper-bounded by a constant (Lemma 4.2).

For a function f and a set of indices I, we define the *projection* of f to I to be the function over $\{0,1\}^n$ whose value depends only on the variables in I and its value at x_I is the expectation of f over all the possible values of variables outside of I, namely $f_I(x) = \mathbf{E}_{y\sim\mathcal{U}}[f(x_I, y_{\bar{I}})]$. We start by establishing several simple properties of projections and influences.

Lemma 6.3. *Let $f : \{0,1\}^n \to \mathbb{R}$ be any function, $i \in [n]$ and $I \subseteq [n]$. Then*

1. *for every I-junta h, $\|f-h\|_2 \geq \|f-f_I\|_2$;*
2. *If $i \in I$ then $(\partial_i f)_I = \partial_i f_I$;*
3. *$\mathsf{Infl}^1_i(f) \leq \|f\|_1$;*
4. *$\mathsf{Infl}^1_i(f_I) \leq \mathsf{Infl}^1_i(f)$;*
5. *$|\hat{f}(\{i\})| \leq \mathsf{Infl}^1_i(f)$;*
6. *[FKV13] for all $j \neq i$, $|\hat{f}(\{i,j\})| = \mathsf{Infl}^1_i(\partial_j f)/2$;*
7. *$\|f-f_I\|_1 \leq \sum_{j\notin I}\mathsf{Infl}^1_i(f)$; for all $J \subseteq [n]$, $\|f_J - f_{I\cap J}\|_1 \leq \sum_{j\in J\setminus I}\mathsf{Infl}^1_j(f_J)$.*

Proof.

1. As is well-known, for any set of m real values $a_1,\ldots,a_m$, the value of $\sum_i(b-a_i)^2$ is minimized when $b = \frac{1}{m}\sum a_i$. Therefore f_I is the I-junta closest (in ℓ_2-norm) to f.
2. For $b \in \{0,1\}$ let f_b be defined as $f_{i\leftarrow b}(x) = f(x_{i\leftarrow b})$. First, observe that if $i \in I$ then we can exchange the restriction and projection operators on f, that is, for every x, $f_{i\leftarrow b,I}(x) = f_{I,i\leftarrow b}(x)$. Now
$$(\partial_i f)_I = (f_{i\leftarrow 1} - f_{i\leftarrow 0})_I = f_{i\leftarrow 1,I} - f_{i\leftarrow 0,I} = f_{I,i\leftarrow 1} - f_{I,i\leftarrow 0} = \partial_i f_I \ .$$

[2] The definition in [BH12] uses the condition $h(x) \leq f(x) \leq \alpha h(x)$ which is equivalent up to scaling the hypothesis by α.

3. $$\mathsf{Infl}^1_i(f) = \mathbf{E}\left[\left|\frac{f(x_{i\leftarrow 1}) - f(x_{i\leftarrow 0})}{2}\right|\right] \leq \mathbf{E}\left[\frac{|f(x_{i\leftarrow 1})| + |f(x_{i\leftarrow 0})|}{2}\right] = \mathbf{E}[|f(x)|] = \|f\|_1.$$

4. Convexity of $|\cdot|$ implies that for every function $g : \{0,1\}^n \to \mathbb{R}$, $\|g_I\|_1 \leq \|g\|_1$. Together with property (2) this implies that
$$\mathsf{Infl}^1_i(f_I) = \mathbf{E}[|\partial_i f_I|]/2 = \mathbf{E}[|(\partial_i f)_I|]/2 \leq \mathbf{E}[|\partial_i f|]/2 = \mathsf{Infl}^1_i(f) \ .$$

5. $$|\hat{f}(\{i\})| = |\mathbf{E}[\partial_i f]|/2 \leq \mathbf{E}[|\partial_i f|]/2 = \mathsf{Infl}^1_i(f) \ .$$

6. $$|\hat{f}(\{i,j\})| =^* \frac{1}{4}|\mathbf{E}[\partial_{i,j} f]| =^{**} \frac{1}{4}\mathbf{E}_{\mathcal{U}}[|\partial_{i,j} f|] = \frac{1}{2}\mathsf{Infl}^1_i(\partial_j f).$$
Here, $(*)$ follows from the basic properties of the Fourier spectrum of partial derivatives (see Sec. 4.1) and $(**)$ is implied by second partial derivatives of a submodular function being always non-positive (see Sec. 2).

7. First,
$$\|f - f_{[n]\setminus\{j\}}\|_1 = \mathbf{E}\left[\left|\frac{f(x_{j\leftarrow 0}) + f(x_{j\leftarrow 1})}{2} - f(x)\right|\right] \leq \mathbf{E}\left[\frac{|f(x_{j\leftarrow 0}) - f(x)|}{2}\right] + \mathbf{E}\left[\frac{|f(x_{j\leftarrow 1}) - f(x)|}{2}\right]$$
$$= \mathbf{E}[|(f(x_{j\leftarrow 1}) - f(x_{j\leftarrow 0})|/2] = \mathbf{E}[|\partial_j(f)|]/2 = \mathsf{Infl}^1_j(f) \ .$$
Together with property (4), this implies that for any $j \notin I$, $\|f_{I\cup\{j\}} - f_I\|_1 \leq \mathsf{Infl}^1_j(f_{I\cup\{j\}}) \leq \mathsf{Infl}^1_j(f)$. By applying this iteratively to all $j \notin I$ and using the triangle inequality we obtain that
$$\|f - f_I\|_1 \leq \sum_{j\notin I} \mathsf{Infl}^1_j(f) \ .$$
To obtain the second part we apply the first part to f_J and obtain
$$\|f_J - f_{I\cap J}\|_1 \leq \sum_{j\notin I} \mathsf{Infl}^1_j(f_J) \ .$$
Observe that for all $j \notin J$, $\mathsf{Infl}^1_j(f_J) = 0$ and hence
$$\sum_{j\notin I} \mathsf{Infl}^1_j(f_J) = \sum_{j\in J\setminus I} \mathsf{Infl}^1_j(f_J) \ .$$

□

We now prove that throwing away variables with small degree-1 or 2 Fourier coefficients does not affect a projection of f to a small set of variables J significantly.

Lemma 6.4. *Let $f : \{0,1\}^n \to \mathbb{R}$ be a real-valued function and let $J \subseteq [n]$. Let*
$$I' = \left\{i \;\middle|\; |\hat{f}(\{i\})| \geq \frac{\epsilon}{2\cdot|J|}\right\} \bigcup \left\{i \;\middle|\; \exists j, |\hat{f}(\{i,j\})| \geq \frac{\epsilon}{2\cdot|J|^2}\right\}$$
and let $I \supseteq I'$. Then $\|f_J - f_{I\cap J}\|_1 \leq \epsilon$.

Proof. By Lem. 6.3(7) we obtain that
$$\|f_J - f_{I\cap J}\|_1 \leq \sum_{i\in J\setminus I} \mathsf{Infl}^1_i(f_J) = \frac{1}{2}\sum_{i\in J\setminus I} \|\partial_i f_J\|_1. \tag{7}$$
We now apply Lem. 6.3(7) to $\partial_i f$ and the empty set projection:
$$\|(\partial_i f)_J - (\partial_i f)_\emptyset\|_1 \leq \sum_{j\in J\setminus\{i\}} \mathsf{Infl}^1_j((\partial_i f)_J) \ . \tag{8}$$

By Lem. 6.3(4,6), $\mathsf{Infl}^1_j((\partial_i f)_J) \le \mathsf{Infl}^1_j(\partial_i f) = 2|\hat{f}(\{i,j\})|$. For $i \notin I$, $|\hat{f}(\{i,j\})| \le \epsilon/(2|J|^2)$. By substituting this into equation (8) we get that

$$\|(\partial_i f)_J - (\partial_i f)_\emptyset\|_1 \le \sum_{j \in J \setminus \{i\}} 2 \cdot \frac{\epsilon}{2 \cdot |J|^2} \le \frac{\epsilon}{|J|}.$$

Now we note that $(\partial_i f)_\emptyset \equiv \mathbf{E}[\partial_i f] = -2\hat{f}(\{i\})$ and (by Lem. 6.3(2)) $(\partial_i f)_J = \partial_i f_J$. This implies that for $i \notin I$,

$$\|\partial_i f_J\|_1 \le \|\partial_i f_J - (\partial_i f)_\emptyset\|_1 + \|(\partial_i f)_\emptyset\|_1 \le \frac{\epsilon}{|J|} + 2|\hat{f}(\{i\})| \le \frac{2\epsilon}{|J|} \ .$$

Substituting this into equation (7) we obtain that

$$\|f_J - f_{I\cap J}\|_1 \le \frac{1}{2} \sum_{i \in J \setminus I} \|\partial_i f_J\|_1 \le \frac{1}{2} \sum_{i \in J \setminus I} \frac{2\epsilon}{|J|} \le \epsilon \ .$$

□

We next bound the number of variables that have large degree-1 or degree-2 Fourier coefficient (a weaker bound is also implied by Parseval's identity).

Lemma 6.5. *Let $f : \{0,1\}^n \to [0,1]$ be a submodular function and $\alpha, \beta > 0$. Let*

$$I = \left\{ i \ \middle|\ |\hat{f}(\{i\})| \ge \alpha \right\} \bigcup \left\{ i \ \middle|\ \exists j, |\hat{f}(\{i,j\})| \ge \beta \right\} \ .$$

Then $|I| \le \frac{2}{\min\{\alpha,\beta\}}$.

Proof. If $i \in I$ then either $|\hat{f}(\{i\})| \ge \alpha$ or $|\hat{f}(\{i,j\})| \ge \beta$ for some $j \ne i$. In the former case, by Lem. 6.3(5), $\mathsf{Infl}^1_i(f) \ge |\hat{f}(\{i\})| \ge \alpha$ and in the latter case, by Lem. 6.3(3,6)

$$\mathsf{Infl}^1_i(f) = \frac{1}{2}\|\partial_i f\|_1 \ge \frac{1}{2}\mathsf{Infl}^1_j(\partial_i f) = |\hat{f}(\{i,j\})| \ge \beta \ .$$

This implies that for all $i \in I$, $\mathsf{Infl}^1_i(f) \ge \min\{\alpha, \beta\}$. By Lemma 4.2, $\mathsf{Infl}^1(f) = \sum_{i \in [n]} \mathsf{Infl}^1_i(f) \le 2$. This gives the claimed bound on $|I|$. □

We are now ready to complete the proof of Theorem 6.2.

Proof of Theorem 6.2. Let $J \subseteq [n]$ be a set of indices of size $s_f(\epsilon/2)$ such that there exists a J-junta g for which $\|f - g\|_2 \le \epsilon/2$. By Lem. 6.3(1), this implies that $\|f - f_J\|_1 \le \|f - f_J\|_2 \le \|f - g\|_2 \le \epsilon/2$. Let

$$I' = \left\{ i \ \middle|\ |\hat{f}(\{i\})| \ge \frac{\epsilon}{4 \cdot s_f(\epsilon/2)} \right\} \bigcup \left\{ i \ \middle|\ \exists j, |\hat{f}(\{i,j\})| \ge \frac{\epsilon}{8 \cdot (s_f(\epsilon/2))^2} \right\} \ .$$

By Lemma 6.4, for any $I \supseteq I'$, $\|f_J - f_{I\cap J}\|_1 \le \epsilon/2$. In particular, it is easy to see that $f_{I\cap J}$ is a submodular $(I \cap J)$-junta. Clearly, $J \cap I \subseteq I$ and $|J \cap I| \le s_f(\epsilon/2)$. By the triangle inequality, $\|f - f_{I\cap J}\|_1 \le \epsilon$.

All we need now is to find a small set of indices $I \supseteq I'$. We simply estimate degree-1 and 2 Fourier coefficients of f to accuracy $\epsilon/(32 \cdot (s_f(\epsilon/2))^2)$ with confidence at least 5/6 using random examples. Let $\tilde{f}(S)$ for $S \subseteq [n]$ of size 1 or 2 denote the obtained estimates. We define

$$I = \left\{ i \ \middle|\ |\tilde{f}(\{i\})| \ge \frac{3\epsilon}{16 \cdot s_f(\epsilon/2)} \right\} \bigcup \left\{ i \ \middle|\ \exists j, |\tilde{f}(\{i,j\})| \ge \frac{3\epsilon}{32 \cdot (s_f(\epsilon/2))^2} \right\} \ .$$

If estimates are within the desired accuracy, then clearly, $I \supseteq I'$. At the same time $I \subseteq I''$, where

$$I'' = \left\{ i \ \middle|\ |\hat{f}(\{i\})| \ge \frac{\epsilon}{8 \cdot s_f(\epsilon/2)} \right\} \bigcup \left\{ i \ \middle|\ \exists j, |\hat{f}(\{i,j\})| \ge \frac{\epsilon}{16 \cdot (s_f(\epsilon/2))^2} \right\} \ .$$

By Lem. 6.5, $|I''| \le 32 \cdot (s_f(\epsilon/2))^2/\epsilon$.

Finally, to bound the running time we observe that, by the standard application of Chernoff bound with the union bound, $O(\log(n) \cdot (s_f(\epsilon/2))^4/\epsilon^2)$ random examples are sufficient to obtain the desired estimates with confidence of 5/6. The estimation of the coefficients can be done in $O(n^2 \log(n) \cdot (s_f(\epsilon/2))^4/\epsilon^2)$ time. □

Our main structural result together with Theorem 6.2 imply that, given random examples of a submodular function f, one can find $\tilde{O}(1/\epsilon^5)$ variables such that there exists a submodular $\tilde{O}(1/\epsilon^2)$-junta over those variables ϵ-close to f.

Corollary 6.6. *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 $\|f - h\|_1 \le \epsilon$. The algorithm runs in time $\tilde{O}(n^2/\epsilon^{10})$ and uses $\tilde{O}(\log(n)/\epsilon^{10})$ examples.*

For general low-influence functions we do not expect to be able to find the influential variables efficiently using random examples alone. For example, Boolean k-juntas have total ℓ_1-influence of at most k but finding the influential variables in $n^{o(k)}$ time is a notoriously hard open problem. However in the special case of monotone functions it is well-known that the influential variables can be found efficiently from random examples alone [Ser04]. The detection of influential variables is based on a simple relationship between ℓ_1-influences of a monotone (and even unate) function and its degree-1 Fourier coefficients.

Lemma 6.7. *Let f be a unate real-valued function. Then for every $i \in [n]$,*

$$|\hat{f}(\{i\})| = \mathsf{Infl}^1_i(f).$$

Proof. By definition, $\mathsf{Infl}^1_i(f) = \mathbf{E}[|\partial_i f|]/2$. For a unate f, $\partial_i f$ is either non-negative for all x or non-positive for all x. Therefore

$$\mathsf{Infl}^1_i(f) = \mathbf{E}[|\partial_i f|]/2 = |\mathbf{E}[\partial_i f]|/2 = |\hat{f}(\{i\})|.$$

□

Therefore to find influential variables it is sufficient to estimate the degree-1 Fourier coefficients (in the same way as in the proof of Thm. 6.2). As an immediate corollary of this observation and Cor. 4.8 we get the following algorithm.

Corollary 6.8. *Let $f : \{0,1\}^n \to [0,1]$ be any 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 $2^{O(\mathsf{Infl}^1(f)/\epsilon^2)}$ such that there exists a function p of Fourier degree $2 \cdot \mathsf{Infl}^1(f)/\epsilon^2$ over variables in I satisfying $\|f - p\|_2 \le \epsilon$. The algorithm runs in time $\tilde{O}(n) \cdot 2^{O(\mathsf{Infl}^1(f)/\epsilon^2)}$ and uses $\log(n) \cdot 2^{O(\mathsf{Infl}^1(f)/\epsilon^2)}$ examples.*

6.3 Proper PAC Learning of Submodular Functions

In this section we use our junta approximation result and the algorithm for finding the influential variables to get a proper learning algorithm for submodular functions. The previous result on approximation by juntas [FKV13] only gives a doubly exponential $2^{2^{O(1/\epsilon^2)}}$ dependence of running time on ϵ. This algorithm also serves as a step in our PMAC learning algorithm.

Theorem 6.9. *There exists an algorithm $\mathcal{A}$ that given $\epsilon > 0$ and access to random uniform examples of any submodular $f : \{0,1\}^n \to [0,1]$, with probability at least $2/3$, outputs a* **submodular** *function h, such that $\|f - h\|_1 \le \epsilon$. Further, h is a J-junta for some J of size $O(1/\epsilon^2 \cdot \log(1/\epsilon))$ variables, $\mathcal{A}$ also returns J and runs in time $\tilde{O}(n^2/\epsilon^{10}) + 2^{\tilde{O}(1/\epsilon^2)}$ and uses $\tilde{O}(\log(n)/\epsilon^{10}) + 2^{\tilde{O}(1/\epsilon^2)}$ random examples.*

The proper learning algorithm.

1. Run the algorithm from Cor. 6.6 to find a set of variables I of size s such that there exists a submodular t-junta g over variables in I satisfying $\|f - g\|_1 \le \epsilon/2$ (with probability at least 5/6).
2. Request m random examples: $(x^1, f(x^1)), (x^2, f(x^2)), \ldots, (x^m, f(x^m))$.
3. FOR every subset $J \subseteq I$ of size t DO
    (a) Solve an LP to find a J-junta $h : \{0,1\}^n \to [0,1]$ that minimizes $\frac{1}{m}\sum_{i\le m} |f(x^i) - h(x^i)|$ with constraints requiring that h be submodular.
    (b) If $\frac{1}{m}\sum_{i\le m} |f(x^i) - h(x^i)| \le 3\epsilon/4$ then return h, J and terminate.
4. Return $h \equiv 0$.

Proof. The specific choices of $s = \tilde{O}(1/\epsilon^5)$ and $t = O(1/\epsilon^2 \cdot \log(1/\epsilon))$ are determined by Cor. 6.6. We choose the number of examples m so as to ensure that, with probability at least 5/6, for every J-junta h such that $J \subseteq I$ and $|J| = t$,

$$\left| \mathbf{E}[|f(x) - h(x)|] - \frac{1}{m} \sum_{i \leq m} |f(x^i) - h(x^i)| \right| \leq \frac{\epsilon}{4} \; . \tag{9}$$

Standard uniform convergence bounds [Vap98] imply that for any fixed set J, using $O(2^t/\epsilon^2 \cdot \log(1/\delta))$ examples will suffice to make sure that equation (9) holds with probability at least $1-\delta$ for all J-juntas with range $[0,1]$. Using the union bound over all $\binom{s}{t} = 2^{\tilde{O}(1/\epsilon^2)}$ subsets of I we get that $m = 2^{\tilde{O}(1/\epsilon^2)}$ will suffice to achieve the desired guarantee.

Now, by Cor. 6.6, there exist $J' \subseteq I$ of size t and a submodular J'-junta g that satisfies $\|f - g\|_1 \leq \epsilon/2$. By equation (9), $\frac{1}{m}\sum_{i \leq m} |f(x^i) - g(x^i)| \leq \|f - g\|_1 + \epsilon/4 \leq 3\epsilon/4$. This implies that when $J = J'$ the solution of LP will be returned as a hypothesis and the algorithm will not reach Step (4) (assuming that Step (1) is successful and equation (9) holds).

For any h returned as a hypothesis, $\frac{1}{m}\sum_{i \leq m} |f(x^i) - h(x^i)| \leq 3\epsilon/4$ and therefore by equation (9), $\|f - h\|_1 \leq \frac{1}{m}\sum_{i \leq m} |f(x^i) - h(x^i)| + \epsilon/4 \leq \epsilon$. This implies that if Step (1) is successful and equation (9) holds then the algorithm will output a hypothesis h with ℓ_1-error of at most ϵ. These conditions hold with probability at least 5/6.

For a t-junta h, the minimization of ℓ_1-error on examples, submodularity and range $[0,1]$ can all be expressed in a linear program with $O(t^2 \cdot 2^t)$ constraints on the values of h at 2^t points. The solution to this LP can be found in time $2^{O(t)}$. Therefore the total running time of Step (3) is $\binom{s}{t} \cdot 2^{O(t)} = 2^{\tilde{O}(1/\epsilon^2)}$. Combining this with the bounds on the running time and the number of examples from Cor. 6.6 we get the claimed bounds. □

6.4 PMAC Learning of Submodular Functions

We now show that approximation by a junta can also be used to obtain a PMAC learning algorithm for submodular functions. Our algorithm is based on a reduction from multiplicative approximation to additive approximation. Specifically we use the additive approximation algorithm (Thm. 6.9) to find a function $g : \{0,1\}^n \to [0,1]$ over a set of variables $J \subseteq [n]$ that has low ℓ_1-error. We then prove that, for at least 1/10 fraction of values $z \in \{0,1\}^J$, g gives a multiplicative approximation to f on at least $1-\epsilon$ fraction of points (z, y) for $y \in \{0,1\}^{\bar{J}}$. This reduces the problem to finding multiplicative approximation to f for values z where the above guarantee does not hold. In other words, we reduce the problem to $\frac{9}{10}2^{|J|}$ instances of the same problem on a subcube of $\{0,1\}^n$ and execute our algorithm recursively for each of those instances. Importantly, this step solves the problem on 1/10-fraction of all the inputs and therefore the depth of the recursion needs to be at most $O(\log(1/\epsilon))$. This makes the total number of executions of this procedure upper bounded by $2^{O(|J| \cdot \log(1/\epsilon))}$.

A crucial property of submodular functions that is needed for this reduction step to work is that when a (non-negative) submodular function f is scaled so that $\|f\|_\infty = 1$, then f equals at least some constant c_1 on at least a constant fraction of inputs. We obtain this property from the following lemma (from [FMV07b]).

Lemma 6.10. *Let $f : \{0,1\}^n \to \mathbb{R}_+$ be a submodular function. Then $\|f\|_1 \geq \frac{1}{4}\|f\|_\infty$.*

We note that this property also holds for XOS functions (with $\frac{1}{2}$ instead of $\frac{1}{4}$; see [Fei06]). Together with Chernoff-Hoeffding's bound, Lemma 6.10 also implies the following lemma.

Lemma 6.11. *There is a constant $c > 0$ such that for any submodular $f : \{0,1\}^n \to \mathbb{R}_+$, $\gamma \in (0,1)$, any integer t, and t points $x^1, x^2, \ldots, x^t$ drawn randomly and uniformly from $\{0,1\}^n$, it holds that*

$$\Pr\left[\left|\frac{1}{t}\sum_{i \in [t]} f(x^i) - \mathbf{E}[f]\right| \geq \gamma \mathbf{E}[f]\right] \leq 2e^{-ct\gamma^2}.$$

Proof. Suppose $\|f\|_\infty = M$, hence $f(x^i)$ are independent random variables in $[0, M]$. By Chernoff-Hoeffding bounds, $\Pr[|\frac{1}{t}\sum_{i=1}^{t} f(x^i) - \mathbf{E}[f]| > \delta M] \leq 2e^{-c'\delta^2 t}$ for some $c' > 0$. We also have $\mathbf{E}[f] = \|f\|_1 \geq \frac{1}{4}M$, hence we can set $\delta = \frac{1}{4}\gamma$ and the lemma follows. □

We now present the details of the algorithm and its analysis.

Theorem 6.12 (Thm. 1.4 restated)**.** *There exists an algorithm $\mathcal{A}$ that given $\gamma, \epsilon \in (0,1]$ and access to random and uniform examples of any submodular function $f : \{0,1\}^n \to \mathbb{R}_+$, with probability at least $2/3$, outputs a function h which multiplicatively $(1+\gamma, \epsilon)$-approximates 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.*

Proof. The algorithm $\mathcal{A}$ relies on the reduction we outlined above. Let $\mathcal{A}'(k)$ denote the execution of the learning algorithm at the k-th level of recursion. $\mathcal{A}$ executes $\mathcal{A}'(0)$ and $\mathcal{A}'(k)$ is the following algorithm.

1. If $k \geq 10\log(1/\epsilon)$ then $\mathcal{A}'(k)$ **returns** the hypothesis $h \equiv 0$.

2. Otherwise, the algorithm estimates $\mathbf{E}[f]$ to within a multiplicative factor of $6/5$ (with probability at least $1-\delta$ for δ to be defined later). Let μ denote the obtained estimate (that is, $\mathbf{E}[f] \leq \mu \leq \frac{6}{5}\mathbf{E}[f]$). If $\mu = 0$ the algorithm **returns** $h \equiv 0$. Otherwise, we define the function $f' = f/(4\mu)$. By Lemma 6.10 we know that $\|f'\|_\infty = \|f\|_\infty/(4\mu) \leq 4\mathbf{E}[f]/(4\mu) \leq 1$.

3. We run our ℓ_1-learning algorithm from Theorem 6.9 on random examples of f' with accuracy $\epsilon' = \gamma\epsilon/2400$ and confidence $1-\delta$ (using the standard confidence boosting technique). Let g be the hypothesis output by the algorithm and J be the set of indices of variables it depends on. We treat g as a function on $\{0,1\}^J$.

4. We define the output hypothesis h by defining for every $z \in \{0,1\}^J$ a function $h_z : \{0,1\}^{\bar{J}} \to \mathbb{R}_+$ and then setting $h(x) = h_{x_J}(x_{\bar{J}})$. For $z \in \{0,1\}^J$, h_z is defined as follows:

   (a) If $g(z) \geq 1/20$ and $\mathbf{E}_{y\in\{0,1\}^{\bar{J}}}[|g(z) - f'(z,y)|] \leq 20\epsilon'$ then define $h_z \equiv (4\mu)(1+\gamma/60)g(z)$

   (b) Otherwise, execute $\mathcal{A}'(k+1)$ on function f_z over $\{0,1\}^{\bar{J}}$ defined as $f_z(y) = f(z,y)$. Let h_z be the output of this execution.

   To simulate random examples of f_z we draw random examples of f until an example (x,ℓ) is obtained such that $x_J = z$. Note that we cannot find $\mathbf{E}_{y\in\{0,1\}^{\bar{J}}}[|g(z) - f'(z,y)|]$ exactly but an estimate within $\epsilon'/2$ with probability $1-\delta$ would suffice (with minor adjustments in the constants).

   $\mathcal{A}'(k)$ **returns** the hypothesis h.

We now prove the correctness of the algorithm under the assumption that random estimations and executions of the algorithm from Theorem 6.9 are successful. First we observe that if the condition in step (4a) holds then h_z multiplicatively $(1+\gamma, \epsilon/2)$-approximates f_z over the uniform distribution on $\{0,1\}^{\bar{J}}$. By Markov's inequality, the condition $\mathbf{E}_{y\in\{0,1\}^{\bar{J}}}[|g(z) - f'(z,y)|] \leq 20\epsilon' = \epsilon\gamma/120$ implies that

$$\Pr_{y\in\{0,1\}^{\bar{J}}}[|g(z) - f'(z,y)| \geq \gamma/60] \leq \epsilon/2 \ .$$

This means that on all but $\epsilon/2$ fraction of points y, $|g(z) - f'(z,y)| \leq \gamma/60$. On those points $g(z) + \gamma/60 \geq f'(z,y)$. In addition, $f'(z,y) \geq g(z) - \gamma/60 \geq 1/20 - \gamma/60 \geq 1/30$ and therefore $g(z) + \gamma/60 \leq f'(z,y) + \gamma/30 \leq (1+\gamma)f'(z,y)$. This implies that $g(z) + \gamma/60$ multiplicatively $(1+\gamma, \epsilon/2)$-approximates $f'(z,y)$. By our definition $h_z \equiv (4\mu)(1+\gamma/60)g(z)$ and $f_z(y) = (4\mu)f'(z,y)$.

Now we observe that we can partition the domain $\{0,1\}^n$ into two sets of points:

1. Set G where either $\mu = \mathbf{E}[f] = 0$ or the value output by the hypothesis is $(4\mu)(1+\gamma/60)g(z)$, where g is returned by one of the invocations of the additive approximation algorithm;

2. The set of points where the recursion reached depth $k > 10\log(1/\epsilon)$.

By the construction, the points in G can be divided into disjoint subcubes such that in each of them the conditional probability that the hypothesis we output does not satisfy the multiplicative guarantee is at most $\epsilon/2$. Therefore the hypothesis does not satisfy the multiplicative guarantee on at most fraction $\epsilon/2$ of the points in G.

To finish the proof of correctness it suffices to show that $\Pr[x \notin G] \leq \epsilon/2$. To establish this we prove that the fraction of points on which $\mathcal{A}'(k)$ is invoked is at most $(9/10)^{-k}$. We prove this by induction (with $k = 0$ being obvious). For any $k < 10\log(1/\epsilon)$ if $\mu = 0$ then $\mathcal{A}'(k+1)$ is not invoked. Otherwise, we know that g satisfies

$$\mathbf{E}[|f'(x) - g(x_J)|] = \mathbf{E}_{z\sim\{0,1\}^J}\left[\mathbf{E}_{y\in\{0,1\}^{\bar{J}}}[|f'(z,y) - g(z)|]\right] \leq \epsilon' \ .$$

Therefore by Markov's inequality,

$$\Pr_{z\sim\{0,1\}^J}\left[\mathbf{E}_{y\in\{0,1\}^{\bar{J}}}[|f'(z,y)-g(z)|] \geq 20\epsilon'\right] \leq 1/20. \tag{10}$$

We also know that

$$\mathbf{E}_{z\sim\{0,1\}^J}[g(z)] \geq \mathbf{E}[f'(x)] - \epsilon' \geq \frac{\mathbf{E}[f(x)]}{4\mu} - \epsilon' \geq \frac{5}{24} - \frac{\epsilon\gamma}{2400} > \frac{1}{5}\ .$$

At the same time g has range $[0,1]$ and hence

$$\mathbf{E}_{z\sim\{0,1\}^J}[g(z)] \leq \frac{1}{20}\Pr_{z\sim\{0,1\}^J}[g(z) < 1/20] + \Pr_{z\sim\{0,1\}^J}[g(z) \geq 1/20]\ .$$

This implies that $\Pr_{x\sim\{0,1\}^J}[g(z) \geq 1/20] \geq 1/5 - 1/20 = 3/20$. Together with equation (10) this implies that the fraction of z's for which both $g(z) \geq 1/20$ and $\mathbf{E}_{y\in\{0,1\}^{\bar{J}}}[|f'(z,y)-g(z)|] \leq 20\epsilon'$ hold is at least $3/20 - 1/20 = 1/10$. This implies that $\mathcal{A}'(k+1)$ will be invoked on at most 9/10-fraction of inputs on which $\mathcal{A}'(k)$ was invoked, proving the inductive claim.

We now also establish the bounds on the running time and sample complexity of this algorithm. Let t be the bound on the size of J in any of the executions of the additive approximation algorithm (Thm. 6.9). Note that the size of junta does not depend on n or the confidence parameter δ and therefore is $\tilde{O}(1/(\epsilon\gamma)^2)$. Let r be the total number of times $\mathcal{A}'$ is executed. From our correctness analysis we can conclude that $r \leq (2^t)^{10\log(1/\epsilon)} = 2^{\tilde{O}(1/(\epsilon\gamma)^2)}$. It is sufficient to set $\delta = 1/(9r)$ to ensure that the total probability of failure is at most 1/3. By Lemma 6.11 and Thm. 6.9 it is easy to see that each execution of $\mathcal{A}'(k)$ 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 (of f restricted to a subcube). Simulating a random example for the execution of $\mathcal{A}'(k)$ requires filtering examples which have $t\cdot k$ variables set to a specific value. This means that simulating $\mathcal{A}'(k)$ requires $2^{jk} = 2^{\tilde{O}(1/(\epsilon\gamma)^2)}$ times more examples of f than the number of examples required by $\mathcal{A}'(k)$. Altogether all r executions run in $\tilde{O}(n^2)\cdot 2^{\tilde{O}(1/(\epsilon\gamma)^2)}$ and use $\log(n)\cdot 2^{\tilde{O}(1/(\epsilon\gamma)^2)}$ examples. Note that we made the standard assumption that manipulating values of f takes $O(1)$ time. □

6.5 Learning of Low-Sensitivity Functions

We now show that using variants of well-known techniques we can obtain close-to-optimal PAC learning algorithms for real-valued functions of low total ℓ_1-influence. We start by proving a generalization of Theorem 1.5.

Theorem 6.13 (subsumes Thm. 1.5)**.** *Let $\mathcal{C}_a^+$ be the set of all unate functions with range in $[0,1]$ and total ℓ_1-influence of at most a. There exists an algorithm $\mathcal{A}$ that given $\epsilon > 0$ and access to random uniform examples of any $f \in \mathcal{C}_a^+$, with probability at least $2/3$, outputs a function h, such that $\|f-h\|_2 \leq \epsilon$. Further, $\mathcal{A}$ runs in time $\tilde{O}(n)\cdot 2^{O(a^2/\epsilon^4)}$ and uses $\log n \cdot 2^{O(a^2/\epsilon^4)}$ examples.*

Proof. Using Corollary 6.8 we can find a set of variables I of size $|I| = 2^{O(a/\epsilon^2)}$ such that there exists a function p of Fourier degree $d = 2a/\epsilon^2$ over variables in I satisfying $\|f-p\|_2 \leq \epsilon/2$. This function is a linear combination of $m = 2^{O(a^2/\epsilon^4)}$ parities. Standard uniform convergence bounds [Vap98, BM02] imply that by using $t = O(m/\epsilon^2)$ random samples and then solving the least squares regression over all m parities, (with probability $\geq 5/6$) we will get a function h such that $\|f-h\|_2 \leq \|f-p\|_2 + \epsilon/2 \leq \epsilon$. This step requires $n\cdot \mathrm{poly}(m/\epsilon) = n\cdot 2^{O(a^2/\epsilon^4)}$ time. □

XOS functions have total ℓ_1-influence of at most 1 and are monotone. Therefore as an immediate corollary of Thm. 6.13 we obtain Theorem 1.5.

It is easy to see that the algorithm for learning XOS functions returns a function that is a $2^{O(1/\epsilon^2)}$-junta. In addition, as we have noted, Lemma 6.10 also holds for XOS functions. Therefore using essentially the same reduction that we used in Theorem 6.12 we can obtain a PMAC learning algorithm for XOS functions with the following guarantees.

Corollary 6.14. *There exists an algorithm $\mathcal{A}$ that given $\gamma, \epsilon \in (0,1]$ and access to random and uniform examples of any XOS function $f : \{0,1\}^n \to \mathbb{R}_+$, with probability at least $2/3$, outputs a function h which multiplicatively $(1+\gamma,\epsilon)$-approximates f (over the uniform distribution). Further, $\mathcal{A}$ runs in time $\tilde{O}(n)\cdot 2^{2^{\tilde{O}(1/(\epsilon\gamma)^2)}}$ and uses $\log(n)\cdot 2^{2^{\tilde{O}(1/(\epsilon\gamma)^2)}}$ examples.*

7 Applications to Testing and Agnostic Learning

7.1 Testing

As is well-known, proper learning with some additional properties (the learner always returns a submodular hypothesis even if the target is not, or we can verify efficiently whether the hypothesis is submodular) also implies testing from random examples with essentially the same bounds on the running time and the number of examples (in the context of Boolean functions this was first observed in [GGR98]). Considering our proper learning algorithm (Theorem 6.9), we obtain the following result.

Corollary 7.1. *There is a testing algorithm that given $\epsilon > 0$ and access to random examples of a function $f : \{0,1\}^n \to [0,1]$, runs in time $\tilde{O}(n^2/\epsilon^{10}) + 2^{\tilde{O}(1/\epsilon^2)}$ and uses $\tilde{O}(\log(n)/\epsilon^{10}) + 2^{\tilde{O}(1/\epsilon^2)}$ random examples and distinguishes with probability at least $2/3$ the following two cases:*

1. *f is submodular;*
2. *for any submodular function h, $\|f - h\|_1 > \epsilon$.*

Next, we consider the value query oracle model, where we can query $f(x)$ for any $x \in \{-1,1\}^n$. Here, we can improve the number of queries from $2^{\tilde{O}(1/\epsilon^2)}$ to $2^{\tilde{O}(1/\epsilon)}$, by plugging in the submodularity tester from [SV11]. The testing algorithm in [SV11] provides the following guarantee: For $f : \{-1,1\}^n \to \mathbb{R}$, it queries $1/\epsilon^{O(\sqrt{n}\log n)}$ values of f and

- If f is submodular, it returns YES.
- If f is ϵ-far *in Hamming distance* (i.e., for any submodular h, $\Pr[f(x) \neq h(x)] > \epsilon$), then it returns NO.

We observe that this is a stronger notion of testing than testing in ℓ_1-distance: If f is ϵ-far from submodular in ℓ_1-distance, it is also at least ϵ-far in Hamming distance. Hence we can use this tester as a building block in our algorithm.

To obtain a tester in ℓ_1-distance, we can apply the techniques from above to reduce dimension to $O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$ and apply the testing algorithm of Seshadhri and Vondrák [SV11] in this setting. The testing algorithm works as follows.

The testing algorithm.

1. Run the algorithm from Cor. 6.6 to find a set of variables I of size $|I| = \mathrm{poly}(1/\epsilon)$ such that if f is submodular then the projected function $g = f_I$ is also submodular and satisfies $\|f - g\|_1 \leq \epsilon/4$ (with high probability).
2. Run Algorithm 3.5 on function g to further reduce the set of variables to $J \subset I$, $|J| = O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$, such that if f is submodular then the function $h = g_J = f_J$ satisfies $\|f - h\|_1 \leq \epsilon/2$ w.h.p.
3. Let $y^{(1)}, \ldots, y^{(m)}$ be uniformly random samples in $\{0,1\}^{\bar{J}}$ and define $\tilde{h}(x) = \frac{1}{m}\sum_{i=1}^{m} f(x_J, y^{(i)})$ (an approximation to h) where $m = \mathrm{poly}(1/\epsilon)$. Clearly we can simulate a value query to $\tilde{h}$ by m value queries to f.
4. Estimate the distance $\|f - \tilde{h}\|_1$ by taking $\mathrm{poly}(1/\epsilon)$ samples; if $\|f - \tilde{h}\|_1 > 3\epsilon/4$ then answer NO.
5. Run the submodularity tester of [SV11] on function $\tilde{h}$ with parameter $\epsilon/4$ and return its answer.

Theorem 7.2. *The testing algorithm above runs in time $\mathrm{poly}(n, 1/\epsilon) + 2^{\tilde{O}(1/\epsilon)}$, uses $\mathrm{poly}(1/\epsilon)\log n + 2^{\tilde{O}(1/\epsilon)}$ queries to f, and distinguishes with probability at least $2/3$ the following two cases:*

1. *f is submodular;*
2. *for any submodular function h, $\|f - h\|_1 > \epsilon$.*

Proof. Provided that the input function f is submodular, the algorithm from Corollary 6.6 and Algorithm 3.5 successfully finds a subset of variables J, $|J| = O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$ such that $h = f_J$ is $\epsilon/2$-close to f (in ℓ_1). We define the function $\tilde{h}$ by averaging $\mathrm{poly}(1/\epsilon)$ functions $f(x_J, y^{(i)})$ where $y^{(i)}$ are random values in $\{0,1\}^{\bar{J}}$. By Chernoff bounds, $\tilde{h}$ is within a $\mathrm{poly}(\epsilon)$ pointwise error of the true projection $h = f_J$. Since $\|h - f\|_1 \le \epsilon/2$, we will obtain with high probability $\|\tilde{h} - f\|_1 \le 3\epsilon/4$ and the tester will pass Step 4. Moreover, note that $\tilde{h}$ is a submodular function (with probability 1), since each of the functions $f(x_J, y^{(i)})$ is submodular. Finally, in Step 5, we use the submodularity tester from [SV11] which will confirm that $\tilde{h}$ is a submodular function and answer YES.

Conversely, if the input function f is ϵ-far from submodular, it cannot be the case that the projected function $h = f_J$ is simultaneously $\epsilon/2$-close to submodular and also $\epsilon/2$-close to f. Therefore, either the test in Step 4 fails because $\tilde{h}$ is far from f, or the tester in Step 5 fails because $\tilde{h}$ is far from submodularity. Either way, we answer NO with high probability.

The time and query complexity of the tester can be analyzed as follows. Step 1 runs in time $\tilde{O}(n^2/\epsilon^{10})$, and uses $\tilde{O}(\log(n)/\epsilon^{10})$ queries (in this case random examples). Step 2 runs in time $\mathrm{poly}(1/\epsilon)$ and uses $\mathrm{poly}(1/\epsilon)$ queries, since it is essentially a greedy algorithm on $\mathrm{poly}(1/\epsilon)$ variables. We need to estimate the values of the multilinear extension $F(x)$ within a $\mathrm{poly}(\epsilon)$ additive error, which can be done with $\mathrm{poly}(1/\epsilon)$ queries. In Step 3, we generate $\mathrm{poly}(1/\epsilon)$ random samples in $\{0,1\}^{\bar{J}}$, which takes $n \cdot \mathrm{poly}(1/\epsilon)$ time. In Step 4, we estimate the distance $\|\tilde{h} - f\|_1$ using $\mathrm{poly}(1/\epsilon)$ queries to f and $\tilde{h}$; each query to $\tilde{h}$ can be simulated by $\mathrm{poly}(1/\epsilon)$ queries to f. Finally, in Step 5 we run the submodularity tester of [SV11] in dimension $n' = O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$. The running time and query complexity of this tester is $1/\epsilon^{O(\sqrt{n'}\log n')} = 2^{\tilde{O}(1/\epsilon)}$. Again, we are simulating each query to $\tilde{h}$ by $\mathrm{poly}(1/\epsilon)$ queries to f, which is absorbed in the $2^{\tilde{O}(1/\epsilon)}$ query complexity. □

7.2 Agnostic Learning

We now give some applications to agnostic learning [KSS94, Hau92] with ℓ_1-error. Our result generalizes those first obtained for submodular functions in [CKKL12] and in addition uses our junta approximation result to reduce the sample complexity from $n^{O(1/\epsilon^2)}$ to $\log n \cdot 2^{O(1/\epsilon^4)}$.

We start with a brief review of the agnostic model. 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 7.3 (Agnostic learning with ℓ_1-error)**.** *Let $\mathcal{F}$ be a class of real-valued functions on $\{0,1\}^n$ and let D be any fixed distribution on $\{0,1\}^n$. For any distribution P over $\{0,1\}^n \times [0,1]$, let opt$(P,\mathcal{F})$ be defined as:*

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

An algorithm $\mathcal{A}$, is said to agnostically learn $\mathcal{F}$ on D if for every $\epsilon > 0$ and any distribution P on $\{0,1\}^n \times [0,1]$ such that the marginal of P on $\{0,1\}^n$ is D, given access to random independent examples drawn from P, with probability at least $\frac{2}{3}$, $\mathcal{A}$ outputs a hypothesis h such that

$$\mathbf{E}_{(x,\ell)\sim P}[|h(x) - \ell|] \le opt(P,\mathcal{F}) + \epsilon.$$

We now describe our agnostic learning algorithm for functions with low total ℓ_1-influence. The algorithm is essentially a Least-Absolute-Error LP (or ℓ_1-regression) over low-degree monomials with an additional constraint on ℓ_1-norm of the coefficient vector of the polynomial (which is also its spectral ℓ_1-norm). The sample complexity analysis of this algorithm is based on the following uniform convergence result of Kakade, Sridharan and Tewari [KST08] which we include here for completeness.

Theorem 7.4 ([KST08])**.** *For $N, W, R > 0$, Let $\mathcal{B}_1^N(W) = \{w \in \mathbb{R}^N \mid \|w\|_1 \le W\}$ and $\mathcal{B}_\infty^N(R) = \{z \in \mathbb{R}^N \mid \|z\|_\infty \le R\}$. Let P be a distribution over $\mathcal{B}_\infty^N(R) \times \mathbb{R}$. Let $\{(z^1,y^1),\ldots,(z^t,y^t)\}$ be a set of t i.i.d. samples from P. Then, with probability at least $1-\delta$ over the choice of samples, it holds that:*

$$\forall w \in \mathcal{B}_1^N(W),\ \left|\mathbf{E}_{(z,y)\sim P}[|\langle z,w\rangle - y|] - \frac{1}{t}\sum_{i=1}^{t}|\langle z^i,w\rangle - y^i|\right| \le 4\cdot WR\cdot\sqrt{\frac{\log(N/\delta)}{t}}.$$

Theorem 7.5. *Let $\mathcal{C}_a$ be the class of all functions with range in $[0,1]$ and total ℓ_1-influence of at most a. There exists an algorithm that learns $\mathcal{C}_a$ agnostically with ℓ_1-error, runs in time $n^{O(a/\epsilon^2)}$ and uses $\log(n)\cdot 2^{O(a^2/\epsilon^4)}$ examples.*

Proof. Let P be any distribution over $\{0,1\}^n \times [0,1]$ whose marginal distribution over $\{0,1\}^n$ is uniform and let $g^* \in \mathcal{C}_a$ be such that $\mathbf{E}_{(x,\ell)\sim P}[|g^*(x) - \ell|] = \min_{g\in\mathcal{C}_a}\{\mathbf{E}_{(x,\ell)\sim P}[|g(x) - \ell|]\} = \Delta$. By Corollary 4.8, we know that there exists a function p of Fourier degree $d = O(a/\epsilon^2)$ such that $\|p - g^*\|_1 \leq \epsilon/2$ and $W = \|\hat{p}\|_1 = 2^{O(d^2)}$.

We draw t examples $\{(x^i, \ell^i)\}_{i\leq t}$ where t is chosen so as to ensure that, with probability $\geq 5/6$, the maximum of

$$\left|\mathbf{E}_P[|p'(x) - \ell|] - \frac{1}{t}\sum_{i\leq t}|p'(x^i) - \ell^i|\right|$$

taken over all functions p' of Fourier degree d and spectral ℓ_1-norm $\leq W$ is at most $\epsilon/4$.

Now we formulate an LP over coefficients $\{\alpha_S\}_{S\subseteq[n],\ |S|\leq d}$ for all parities of degree at most d that minimizes

$$\sum_{i\leq t}\left|\sum_{S\subseteq[n],\ |S|\leq d}\alpha_S\chi_S(x^i) - \ell^i\right|$$

subject to $\sum_{S\subseteq[n],\ |S|\leq d}|\alpha_S| \leq W$. Let $p'(x)$ be the function obtained by solving this LP. By the definition of our LP,

$$\sum_{i\leq t}|p'(x^i) - \ell^i| \leq \sum_{i\leq t}|p(x^i) - \ell^i| \ . \tag{11}$$

Both p and p' are functions of Fourier degree d and spectral ℓ_1-norm $\leq W$ and therefore by our choice of t, with probability at least $2/3$,

$$\mathbf{E}_P[|p(x) - \ell|] \geq \frac{1}{t}\sum_{i\leq t}|p(x^i) - \ell^i| - \epsilon/4$$

and

$$\mathbf{E}_P[|p'(x) - \ell|] \leq \frac{1}{t}\sum_{i\leq t}|p'(x^i) - \ell^i| + \epsilon/4 \ .$$

This implies that

$$\mathbf{E}_P[|p'(x) - \ell|] - \mathbf{E}_P[|p(x) - \ell|] \leq \frac{1}{t}\sum_{i\leq t}|p'(x^i) - \ell^i| - \frac{1}{t}\sum_{i\leq t}|p(x^i) - \ell^i| + \epsilon/2.$$

By combining this with eq. (11), we get that $\mathbf{E}_P[|p'(x) - \ell|] \leq \mathbf{E}_P[|p(x) - \ell|] + \epsilon/2$ and hence our hypothesis $h(x) = p'(x)$ satisfies

$$\begin{aligned}\mathbf{E}_P[|p'(x) - \ell|] &\leq \mathbf{E}_P[|p(x) - \ell|] + \epsilon/2\\ &\leq \mathbf{E}_P[|g^*(x) - \ell|] + \|g^* - p\|_1 + \epsilon/2 \leq \mathbf{E}_P[|g^*(x) - \ell|] + \epsilon = \Delta + \epsilon.\end{aligned}$$

Finally, to bound t we use Thm. 7.4. Note that the dimension $N = |\{S \mid S \subseteq [n], |S| \leq d\}| \leq n^d$, ℓ_1 constraint on the sum of coefficients is W and $R = 1$ since the range of each parity function is $\{-1,1\}$. This implies that taking $t = O(W^2 \log m/\epsilon^2) = 2^{O(d)}\log n$ examples will ensure uniform convergence with error $\epsilon/4$ and confidence $5/6$. The running time of solving the LP is $n^{O(d)}$. □

Given access to value queries one can make agnostic learning more efficient. We first remark that if one is only concerned with squared error, then agnostic learning of all functions with spectral ℓ_1-norm of W can be done in time $\mathrm{poly}(n, W, 1/\epsilon)$ using the algorithm of Kushilevitz and Mansour [KM93] (see [FKV13] for details). Achieving agnostic learning for ℓ_1-error is substantially more involved. This problem was solved by Gopalan, Kalai and Klivans [GKK08] who proved the following theorem.

Theorem 7.6 (Sparse ℓ_1-regression [GKK08]). *For $W > 0$, we define $\mathcal{C}_W$ as $\{p(x) \mid \|\hat{p}\|_1 \leq W\}$. There exists an algorithm $\mathcal{A}$ that given $\epsilon > 0$ and access to value queries for any real-valued $f : \{0,1\}^n \to [-1,1]$, with probability at least $2/3$, outputs a function h, such that $\|f - h\|_1 \leq \Delta + \epsilon$, where $\Delta = \min_{p\in\mathcal{C}_W}\{\|f - p\|_1\}$. Further, $\mathcal{A}$ runs in time* $\mathrm{poly}(n, W, 1/\epsilon)$

By Corollary 4.8, we know that every function with range in $[0,1]$ and total ℓ_1-influence of at most a can be ϵ-approximated in ℓ_1 norm by a function of spectral ℓ_1-norm $2^{O(a^2/\epsilon^4)}$. Together with Theorem 7.6 this implies the existence of the following learning algorithm.

Theorem 7.7. *Let $\mathcal{C}_a$ be the class of all functions with range in $[0,1]$ and total ℓ_1-influence of at most a. There exists an algorithm that, given access to value query oracle, learns $\mathcal{C}_a$ agnostically with ℓ_1-error in $\mathrm{poly}(n) \cdot 2^{O(a^2/\epsilon^4)}$ time.*

We remark that for the special case of submodular functions, a slightly faster ($\mathrm{poly}(n) \cdot 2^{O(1/\epsilon^2)}$-time) and attribute-efficient algorithm was given in [FKV13].

8 Discussion and Open Problems

Our paper essentially resolves the question of additive approximation by juntas for submodular functions and multiplicative approximation by juntas for monotone submodular functions. However many natural questions and gaps in our bounds still remain. The most obvious one is whether there exists a multiplicative approximation junta for non-monotone submodular functions similar to the monotone case (of size $\mathrm{poly}(1/\gamma, \log(1/\epsilon))$ as in Theorem 1.2). We note that the existence of a multiplicative $(1+\gamma, \epsilon)$-approximation junta of size $2^{\mathrm{poly}(1/\gamma, 1/\epsilon)}$ follows from our PMAC-learning algorithm for non-monotone submodular functions (Section 6.4). It is an interesting question whether there is indeed a significant gap between monotone and non-monotone submodular functions or not. Another natural question is whether every XOS function can be multiplicatively $(1+\gamma, \epsilon)$-approximated by a junta of exponential in $1/\epsilon$ and $1/\gamma$ size.

It would also be interesting to understand under what distributional assumptions (besides uniform/product) such strong approximation-by-junta results hold. Algorithmically, it is interesting whether a junta of close to optimal size (that is, $\tilde{O}(1/\epsilon^2)$) can be found in polynomial time given random examples alone. Our algorithm in Theorem 6.2 only finds a larger $\tilde{O}(1/\epsilon^5)$-junta in polynomial time.

Acknowledgements

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

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.

[Bec75] W. Beckner. Inequalities in Fourier analysis. *Ann. of Math. (2)*, 102(1):159–182, 1975.

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

[BLB03] Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In *Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, pages 208–240, 2003.

[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.

[BM02] P. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *JMLR*, 3:463–482, 2002.

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

[Bon70] A. Bonami. Étude des coefficients de Fourier des fonctions de $L^p(G)$. *Ann. Inst. Fourier (Grenoble)*, 20(fasc. 2):335–402 (1971), 1970.

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

[Bou02] Jean Bourgain. On the distribution of the fourier spectrum of boolean functions. *Israel Journal of Mathematics*, 131(1):269–276, 2002.

[CKK$^+$06] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity*, 15(2):94–114, 2006.

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

[DF05] I. Dinur and E. Friedgut. Lecture notes for analytical methods in combinatorics and computer-science (lect 5). Available at `http://www.cs.huji.ac.il/~analyt/`, 2005.

[DFKO06] I. Dinur, E. Friedgut, G. Kindler, and R. O'Donnell. On the Fourier tails of bounded functions over the discrete cube. In *STOC*, pages 437–446, 2006.

[DS05] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162, 2005.

[DS06] S. Dobzinski and M. Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *SODA*, pages 1064–1073, 2006.

[DS09] I. Diakonikolas and R. Servedio. Improved approximation of linear threshold functions. In *CCC*, pages 161–172, 2009.

[Edm70] Jack Edmonds. Matroids, submodular functions and certain polyhedra. *Combinatorial Structures and Their Applications*, pages 69–87, 1970.

[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.

[FK14] Vitaly Feldman and Pravesh Kothari. Learning coverage functions and private release of marginals. In *COLT*, pages 679–702, 2014.

[FKN02] E. Friedgut, G. Kalai, and A. Naor. Boolean functions whose Fourier transform is concentrated on the first two levels. *Adv. in Appl. Math*, 29, 2002.

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

[FMV07a] U. Feige, V. Mirrokni, and J. Vondrák. Maximizing non-monotone submodular functions. pages 461–471, 2007.

[FMV07b] 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.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.

[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.

[GKS05] C. Guestrin, A. Krause, and A. Singh. Near-optimal sensor placements in gaussian processes. In *ICML*, pages 265–272, 2005.

[GMR12] P. Gopalan, R. Meka, and O. Reingold. DNF sparsification and a faster deterministic counting algorithm. In *CCC*, pages 126–135, 2012.

[GV06] M. Goemans and J. Vondrák. Covering minimum spanning trees of random subgraphs. *Random Struct. Algorithms*, 29(3):257–276, 2006.

[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.

[KGGK06] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Near-optimal sensor placements: maximizing information while minimizing communication cost. In *IPSN*, pages 2–10, 2006.

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

[KM93] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

[Kol01] Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

[KR06] R. Krauthgamer and Y. Rabani. Improved lower bounds for embeddings into L1. In *SODA*, pages 1010–1017, 2006.

[KR08] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2-ϵ;. *JCSS*, 74(3):335–349, May 2008.

[KSG08] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *JMLR*, 9:235–284, 2008.

[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.

[Man95] Y. Mansour. An $O(n^{\log \log n})$ learning algorithm for DNF under the uniform distribution. *Journal of Computer and System Sciences*, 50:543–550, 1995.

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

[NS92] N. Nisan and M. Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 4:462–467, 1992.

[OS07] R. O'Donnell and R. Servedio. Learning monotone decision trees in polynomial time. *SIAM J. Comput.*, 37(3):827–844, 2007.

[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.

[Ser04] R. Servedio. On learning monotone DNF under product distributions. *Information and Computation*, 193(1):57–74, 2004.

[SV11] C. Seshadhri and J. Vondrák. Is submodularity testable? In *Innovations in computer science*, pages 195–210, 2011.

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

[Vap98] V. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, New York, 1998.

[Von08] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, pages 67–74, 2008.

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

A Submodular Functions over General Product Distributions

In this section, we address the question of extending our results to general product distributions on $\{0,1\}^n$. We present our submodular junta approximation result in this more general setting to illustrate how parameters of the distribution affect the statement of the results. Replicating all our results in this general setting is beyond the scope of this work. We note that the extension of the Fourier analysis based tools that we use to this setting is well-known.

Theorem A.1. *Let $\mathcal{D}$ be a product distribution on $\{0,1\}^n$ and let $p_0 = \min_{i\in[n],a\in\{0,1\}} \Pr_{x\sim\mathcal{D}}[x_i = a] > 0$. Then 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 $J \subseteq [n]$, $|J| = O(\frac{1}{p_0\epsilon^2} \log \frac{1}{p_0\epsilon})$, such that $\mathbf{E}_{x\sim\mathcal{D}}[|f(x) - g(x)|^2] \leq \epsilon^2$.*

Note the dependence of the size of the junta on p_0. Let us show first that a factor of $\Omega(1/p_0)$ is necessary.

Proposition A.2. *Let $s \geq 2$ be even and let $\mathcal{D}$ be a product distribution on $\{0,1\}^n$ such that $\Pr_{x\sim\mathcal{D}}[x_i = 1] = p_0 = 1 - (1/2)^{2/s}$ for all $i \in [n]$. Let $f(x) = \min\{\sum_{i\in S} x_i, 1\}$ where $|S| = s$. Then there is no function $g : \{0,1\}^n \to \mathbb{R}$ such that $\mathbf{E}_{x\sim\mathcal{D}}[|f(x) - g(x)|^2] < 1/8$ and g depends on fewer than $s/2$ variables.*

Note that $s = -2/\log_2(1-p_0) = \Omega(1/p_0)$, so the claim is that we need $\Omega(1/p_0)$ variables to approximate f even within a constant ℓ_2 error. To prove this, consider any function g depending on $|J| = s/2$ variables. Variables outside of S do not affect f so we may assume that $J \subset S$. Note that $f(x)$ attains only values 0 and 1. With respect to $\mathcal{D}$, we have $\Pr_{x\sim\mathcal{D}}[f(x) = 1] = \Pr_{x\sim\mathcal{D}}[\exists i \in S; x_i = 1] = 1-(1-p_0)^s = 1-1/4 = 3/4$. Furthermore, $\Pr_{x\sim\mathcal{D}}[\forall i \in J; x_i = 0] = (1 - p_0)^{s/2} = 1/2$. Conditioned on $x_i = 0$ for all $i \in J$, we have $f(x) = 0$ with probability $(1 - p_0)^{s/2} = 1/2$ and $f(x) = 1$ with probability $1/2$. However, $g(x)$ has the same value in all these cases and hence we get $\mathbf{E}[|f(x) - g(x)|^2 \mid x_J = 0] \geq 1/4$ (the best choice is to set $g(x) = 1/2$ whenever $x_J = 0$). Since $\Pr_{x\sim\mathcal{D}}[x_J = 0] = 1/2$, we obtain $\mathbf{E}_{x\sim\mathcal{D}}[|f(x) - g(x)|^2] \geq 1/8$.

Next, we turn to the proof of Theorem A.1. As in the case of uniform distributions, it is sufficient to prove the following lemma which is then iterated to obtain Theorem A.1.

Lemma A.3. *Let $\mathcal{D}$ be a product distribution on $\{0,1\}^n$ such that $p_0 = \min_{i\in[n],a\in\{0,1\}} \Pr_{x\sim\mathcal{D}}[x_i = a] > 0$. 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 $J' \subseteq J$, $|J'| = O(\frac{1}{p_0\epsilon^2} \log \frac{|J|}{\epsilon^2})$, such that $\mathbf{E}_{x\sim\mathcal{D}}[|f(x) - h(x)|^2] \leq \frac{1}{4}\epsilon^2$.*

In the following, we prove this lemma. Again, our proof relies on a greedy procedure to select the significant variables, and the boosting lemma to obtain a high-probability bound on the event that the function is sufficiently Lipschitz in the remaining variables. We need the following general version of the boosting lemma ([GV06], somewhat reformulated here).

Lemma A.4 (non-uniform boosting lemma)**.** *Let $\mathcal{F} \subseteq \{0,1\}^X$ be down-monotone and $\eta \in (0,1)$. Let $\mathcal{D}, \mathcal{D}'$ be product distributions on $\{0,1\}^X$ where $\Pr_{x\sim\mathcal{D}'}[x_i = 0] = (\Pr_{x\sim\mathcal{D}}[x_i = 0])^\eta$ for each $i \in X$. Then*

$$\Pr_{x\sim\mathcal{D}'}[x \in \mathcal{F}] \geq \left(\Pr_{x\sim\mathcal{D}}[x \in \mathcal{F}]\right)^\eta.$$

We choose the significant variables using the following algorithm.

Algorithm A.5. *Given $f : \{0,1\}^J \to [0,1]$ and a product distribution $\mathcal{D}$ on $\{0,1\}^J$, produce a small set of important coordinates J' as follows (for parameters $\alpha, \eta > 0$):*

- *Let $\mathcal{D}_0$ be a product distribution such that $\Pr_{x\sim\mathcal{D}_0}[x_i = 0] = (\Pr_{x\sim\mathcal{D}}[x_i = 0])^\eta$ for each $i \in J$, and $\mathcal{D}_1$ a product distribution such that $\Pr_{x\sim\mathcal{D}_1}[x_i = 1] = (\Pr_{x\sim\mathcal{D}}[x_i = 1])^\eta$ for each $i \in J$.*
- *Set $S = T = \emptyset$.*
- *As long as there is $i \notin S$ such that $\Pr_{x\sim\mathcal{D}_0}[\partial_i f(x \wedge \mathbf{1}_S) > \alpha] > 1/2$, include i in S.* (This step is sufficient for monotone submodular functions.)
- *As long as there is $i \notin T$ such that $\Pr_{x\sim\mathcal{D}_1}[\partial_i f(x \vee \mathbf{1}_{J\setminus T}) < -\alpha] > 1/2$, include i in T.* (This step deals with non-monotone submodular functions.)
- *Return $J' = S \cup T$.*

Note how the algorithm changed from the case of uniform distributions: the criterion for selecting variables is now based on discrete derivatives with respect to a non-uniformly sampled set, according to a distribution derived in a certain way from the target distribution $\mathcal{D}$. The goal of this criterion is to achieve the following guarantee.

Lemma A.6. *With the same notation as above, for any $i \in J \setminus J'$*

$$\Pr_{x\sim\mathcal{D}}[\partial_i f(x \wedge \mathbf{1}_{J'}) > \alpha] \leq (1/2)^{1/\eta}$$

and

$$\Pr_{x\sim\mathcal{D}}[\partial_i f(x \vee \mathbf{1}_{J\setminus J'}) < -\alpha] \leq (1/2)^{1/\eta}.$$

Proof. Directly from Lemma A.4, applied to events on the subcube $\{0,1\}^{J'}$: for a variable that was not selected by the algorithm, we have $\Pr_{x\sim\mathcal{D}_0}[\partial_i f(x \wedge \mathbf{1}_{J'}) > \alpha] \leq 1/2$. Since this is a down-monotone event, and $\Pr_{x\sim\mathcal{D}_0}[x_i = 0] = (\Pr_{x\sim\mathcal{D}}[x_i = 0])^\eta$, Lemma A.4 implies $\Pr_{x\sim\mathcal{D}}[\partial_i f(x \wedge \mathbf{1}_{J'})] \leq (1/2)^{1/\eta}$. Similarly (by flipping the cube to change an up-monotone event into a down-monotone one), we obtain $\Pr_{x\sim\mathcal{D}}[\partial_i f(x \vee \mathbf{1}_{J\setminus J'}) < -\alpha] \leq (1/2)^{1/\eta}$. □

The analysis of the size of set J' is identical to the proof of Lemma 3.6. Compared to Lemma 3.6, the only difference is that we keep track of a random subset of the selected variables, sampled according to the distribution $\mathcal{D}_0$ (or complement of $\mathcal{D}_1$, in the second part of the proof). Each selected variable appears in this random set with probability at least $1 - (1-p_0)^\eta \geq \eta p_0$ and hence contributes at least $\frac{1}{2} p_0 \eta \alpha$ to its expected value. Therefore, we obtain the following.

Lemma A.7. *The number of variables chosen by the procedure above is $|J'| \leq \frac{4}{p_0 \alpha \eta}$.*

The rest of the analysis proceeds similarly to the uniform distribution case. We choose $\eta = 1/\log_2 \frac{16|J|}{\epsilon^2}$ and $\alpha = \frac{1}{16}\epsilon^2$. This ensures that when x is sampled according to $\mathcal{D}$ restricted to J', the probability that $f(x,y)$ is α-Lipschitz in the variables y is at least $1 - 2|J| \cdot (1/2)^{1/\eta} \geq 1 - \frac{1}{8}\epsilon^2$. For those points x where $f(x,y)$ is α-Lipschitz in y, we get by Corollary 3.3 that the variance of f is at most $2\alpha = \frac{1}{8}\epsilon^2$. Therefore, as before we conclude that a junta on the variables indexed by J' approximates f within ℓ_2 error $\frac{1}{2}\epsilon$. The size of the junta is $|J'| = O(\frac{1}{\alpha\eta p_0}) = O(\frac{1}{p_0\epsilon^2}\log\frac{|J|}{\epsilon^2})$, which proves Lemma A.3.