Representation, Approximation and Learning of Submodular Functions Using Low-rank Decision Trees

Vitaly Feldman
IBM Research - Almaden

Pravesh Kothari
University of Texas, Austin*

Jan Vondrák
IBM Research - Almaden

April 2, 2013

Abstract

We study the complexity of approximate representation and learning of submodular functions over the uniform distribution on the Boolean hypercube $\{0,1\}^n$. Our main result is the following structural theorem: any submodular function is ϵ-close in ℓ_2 to a real-valued decision tree (DT) of depth $O(1/\epsilon^2)$. This immediately implies that any submodular function is ϵ-close to a function of at most $2^{O(1/\epsilon^2)}$ variables and has a spectral ℓ_1 norm of $2^{O(1/\epsilon^2)}$. It also implies the closest previous result that states that submodular functions can be approximated by polynomials of degree $O(1/\epsilon^2)$ (Cheraghchi et al., 2012). Our result is proved by constructing an approximation of a submodular function by a DT of rank $4/\epsilon^2$ and a proof that any rank-r DT can be ϵ-approximated by a DT of depth $\frac{5}{2}(r + \log(1/\epsilon))$.

We show that these structural results can be exploited to give an attribute-efficient PAC learning algorithm for submodular functions running in time $\tilde{O}(n^2) \cdot 2^{O(1/\epsilon^4)}$. The best previous algorithm for the problem requires $n^{O(1/\epsilon^2)}$ time and examples (Cheraghchi et al., 2012) but works also in the agnostic setting. In addition, we give improved learning algorithms for a number of related settings.

We also prove that our PAC and agnostic learning algorithms are essentially optimal via two lower bounds: (1) an information-theoretic lower bound of $2^{\Omega(1/\epsilon^{2/3})}$ on the complexity of learning monotone submodular functions in any reasonable model (including learning with value queries); (2) computational lower bound of $n^{\Omega(1/\epsilon^{2/3})}$ based on a reduction to learning of sparse parities with noise, widely-believed to be intractable. These are the first lower bounds for learning of submodular functions over the uniform distribution.

1 Introduction

We study the problem of learning submodular functions and their (approximate) representation. Submodularity, a discrete analog of convexity, has played an essential role in combinatorial optimization (Lovász, 1983). It appears in many important settings including cuts in graphs (Goemans and Williamson, 1995, Queyranne, 1995, Fleischer et al., 2001), rank function of matroids (Edmonds, 1970, Frank, 1997), set covering problems (Feige, 1998), and plant location problems (Cornuejols et al., 1977). 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 (Guestrin et al., 2005, Krause et al., 2006, 2008, Krause and Guestrin, 2011) 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*

*Work done while the author was at IBM Research - Almaden.

functions with the property of diminishing returns (B. Lehmann and Nisan, 2006, Dobzinski et al., 2005, Vondrák, 2008, Papadimitriou et al., 2008, Dughmi et al., 2011).

Wide-spread applications of submodular functions have recently inspired the question of whether and how such functions can be learned from random examples (of an unknown submodular function). The question was first formally considered by Balcan and Harvey (2012) who motivate it by learning of valuations functions. Previously, reconstruction of such functions up to some multiplicative factor from value queries (which allow the learner to ask for the value of the function at any point) was also considered by Goemans et al. (2009). These works have lead to significant attention to several variants of the problem of learning submodular functions (Gupta et al., 2011, Cheraghchi et al., 2012, Badanidiyuru et al., 2012, Balcan et al., 2012, Raskhodnikova and Yaroslavtsev, 2013). We survey the prior work in more detail in Sections 1.1 and 1.2.

In this work we consider the setting in which the learner gets random and uniform examples of an unknown submodular function f and its goal is to find a hypothesis function h which ϵ-approximates f for a given $\epsilon > 0$. The main measures of the approximation error we use are the standard absolute error or ℓ_1-distance, which equals $\mathbf{E}_{x\sim D}[|f(x) - h(x)|]$ and ℓ_2-distance which equals $\sqrt{\mathbf{E}_{x\sim D}[(f(x) - h(x))^2]}$ (and upper-bounds the ℓ_1 norm). This is essentially the PAC model (Valiant, 1984) of learning applied to real-valued functions (as done for example by Haussler (1992) and Kearns et al. (1994)). It is also closely related to learning of probabilistic concepts (which are concepts expressing the probability of the function being 1) in which the goal is to approximate the unknown probabilistic concept in ℓ_1 (Kearns and Schapire, 1994). As follows from the previous work (Balcan and Harvey, 2012), without assumptions on the distribution, learning a submodular function to a constant ℓ_1 error requires an exponential number of random examples. We therefore consider the problem with the distribution restricted to be uniform, a setting widely-studied in the context of learning Boolean functions in the PAC model (e.g. Linial et al. (1993), O'Donnell and Servedio (2007)). This special case is also the focus of several other recent works on learning submodular functions (Gupta et al., 2011, Cheraghchi et al., 2012, Raskhodnikova and Yaroslavtsev, 2013).

1.1 Our Results

We give three types of results on the problem of learning and approximating submodular function over the uniform distribution. First we show that submodular functions can be approximated by decision trees of low-rank. Then we show how such approximation can be exploited for learning. Finally, we show that our learning results are close to the best possible.

Structural results: Our two key structural results can be summarized as follows. The first one shows that every submodular function can be approximated by a decision tree of low *rank*. The *rank* of a decision tree is a classic measure of complexity of decisions trees introduced by Ehrenfeucht and Haussler (1989). One way to define the rank of a decision tree T (denoted by $\mathtt{rank}(T)$) is as the depth of the largest complete binary tree that can be embedded in T (see Section 2 for formal definitions).

Theorem 1.1. *Let $f : \{0,1\}^n \to [0,1]$ be a submodular function and $\epsilon > 0$. There exists a real-valued binary decision tree T of rank at most $4/\epsilon^2$ that approximates f within ℓ_2-error ϵ.*

This result is based on a decomposition technique of Gupta et al. (2011) that shows that a submodular function f can be decomposed into disjoint regions where f is also α-Lipschitz (for some $\alpha > 0$). We prove that this decomposition can be computed by a binary decision tree of rank $2/\alpha$. Our second result is that over the uniform distribution a decision tree of rank r can be ϵ-approximated by a decision tree of depth $O(r + \log(1/\epsilon))$.

Theorem 1.2. *Let T be a binary decision tree of rank r. Then for any integer $d \geq 0$, T truncated at depth $d = \frac{5}{2}(r + \log(1/\epsilon))$ gives a decision tree $T_{\leq d}$ such that, $\Pr_{\mathcal{U}}[T(x) \neq T_{\leq d}(x)] \leq \epsilon$.*

It is well-known (e.g. (Kushilevitz and Mansour, 1993)), that a decision tree of size s (i.e. with s leaves) is ϵ-close to the same decision tree pruned at depth $\log(s/\epsilon)$. It is also well-known that for any decision tree of size s has rank of at most $\log s$. Therefore Theorem 1.2 (strictly) generalizes the size-based pruning. Another implication of this result is that several known algorithms for learning polynomial-size DTs over the uniform distribution (e.g. (Kushilevitz and Mansour, 1993, Gopalan et al., 2008)) can be easily shown to also learn DTs of logarithmic rank (which might have superpolynomial size).

Combining Theorems 1.1 and 1.2 we obtain that submodular functions can be approximated by shallow decision trees and consequently as functions depending on at most $2^{\mathrm{poly}(1/\epsilon)}$ variables.

Corollary 1.3. *Let $f : \{0,1\}^n \to [0,1]$ be a submodular function and $\epsilon > 0$. There exists a binary decision tree T of depth $d = O(1/\epsilon^2)$ with constants in the leaves such that $\|T - f\|_2 \leq \epsilon$. In particular, T depends on at most $2^{O(1/\epsilon^2)}$ variables.*

We remark that it is well-known that a DT of depth d can be written as a polynomial of degree d. This gives a simple combinatorial proof of the low-degree approximation of (Cheraghchi et al., 2012) which is based on an analysis of the noise stability of submodular functions. In addition, in our case the polynomial depends only on $2^{O(1/\epsilon^2)}$ variables, which is not true for the approximating polynomial constructed in (Cheraghchi et al., 2012).

Algorithmic applications: We show that these structural results can be used to obtain a number of new learning algorithms for submodular functions. One of the key issues in applying our approximation by a function of few variables is detecting the $2^{O(1/\epsilon^2)}$ variables that would suffice for approximating a submodular function given random examples alone. While for general functions this probably would not be an efficiently solvable problem, we show that a combination of (1) approximation of submodular functions by low-degree polynomials of low spectral (Fourier) ℓ_1 norm (implied by the DT approximation) and (2) the discrete concavity of submodular functions allow finding the necessary variables by looking at Fourier coefficients of degree at most 2.

Lemma 1.4. *There exists an algorithm that given uniform random examples of values of a submodular function $f : \{0,1\}^n \to [0,1]$, finds a set of $2^{O(1/\epsilon^2)}$ variables J such that there is a function f_J depending only on the variables in J and satisfying $\|f - f_J\|_2 \leq \epsilon$. The algorithm runs in time $n^2 \log(n) \cdot 2^{O(1/\epsilon^2)}$ and uses $\log(n) \cdot 2^{O(1/\epsilon^2)}$ random examples.*

Combining this lemma with Corollary 1.3 and using standard Fourier-based learning techniques, we obtain the following learning result in the PAC model.

Theorem 1.5. *There is an algorithm that given uniform random examples of any submodular function $f : \{0,1\}^n \to [0,1]$, outputs a function h, such that $\|f - h\|_2 \leq \epsilon$. The algorithm runs in time $\tilde{O}(n^2) \cdot 2^{O(1/\epsilon^4)}$ and uses $2^{O(1/\epsilon^4)} \log n$ examples.*

In the language of approximation algorithms, we give the first *efficient polynomial-time approximation scheme* (EPTAS) algorithms for the problem. We note that the best previously known algorithm for learning of submodular functions within ℓ_1-error ϵ runs in time $n^{O(1/\epsilon^2)}$ (Cheraghchi et al., 2012), in other words is a PTAS (this algorithm works also in the agnostic setting).

We also give a faster algorithm for agnostic learning of submodular functions, provided that we have access to value queries (returning $f(x)$ for a given point $x \in \{0,1\}^n$).

Theorem 1.6. *Let $\mathcal{C}_s$ denote the class of all submodular functions from $\{0,1\}^n$ to $[0,1]$. There is an agnostic learning algorithm that given access to value queries for a function $f : \{0,1\}^n \to [0,1]$, outputs a function h such that $\|f - h\|_2 \leq \Delta + \epsilon$, where $\Delta = \min_{g \in \mathcal{C}_s}\{\|f - g\|_2\}$. The algorithm runs in time $\mathrm{poly}(n, 2^{1/\epsilon^2})$ and uses $\mathrm{poly}(\log n, 2^{1/\epsilon^2})$ value queries.*

This algorithm is based on an attribute-efficient version of the Kushilevitz-Mansour algorithm (Kushilevitz and Mansour, 1993) for finding significant Fourier coefficients by Feldman (2007). We also show a different algorithm with the same agnostic guarantee but relative to the ℓ_1-distance (and hence incomparable). In this case the algorithm is based on an attribute-efficient agnostic learning of decision trees which results from agnostic boosting (Kalai and Kanade, 2009, Feldman, 2010) applied to the attribute-efficient algorithm for learning parities (Feldman, 2007).

Finally, we discuss the special case of submodular function with a discrete range $\{0, 1, \ldots, k\}$ studied in a recent work of Raskhodnikova and Yaroslavtsev (2013). We show that an adaptation of our techniques implies that such submodular functions can be *exactly* represented by rank-$2k$ decision trees. This directly leads to new structural results and faster learning algorithms in this setting. A more detailed discussion appears in Section B.

Lower bounds: We prove that an exponential dependence on ϵ is necessary for learning of submodular functions (even monotone ones), in other words, there exists no fully polynomial-time approximation scheme (FPTAS) for the problem.

Theorem 1.7. *PAC-learning monotone submodular functions with range $[0,1]$ within ℓ_1-error of $\epsilon > 0$ requires $2^{\Omega(\epsilon^{-2/3})}$ value queries to f.*

Our proof shows that any function g of t variables can be embedded into a submodular function f_g over $2t$ variables in a way that any approximation of f_g to accuracy $\theta(t^{-3/2})$ would yield a $1/4$ approximation of g. The latter is well known to require $\Omega(2^t)$ random examples (or even value queries). This result implies optimality (up to the constant in the power of ϵ) of our PAC learning algorithms for submodular functions.

Further, we prove that agnostic learning of monotone submodular functions is computationally hard via a reduction from learning sparse parities with noise.

Theorem 1.8. *Agnostic learning of monotone submodular functions with range $[0,1]$ within ℓ_1-error of $\epsilon > 0$ in time $T(n, 1/\epsilon)$ would imply learning of parities of size $\epsilon^{-2/3}$ with noise of rate η in time $\mathrm{poly}(n, \frac{1}{\epsilon(1-2\eta)}) + 2T(n, \frac{c}{\epsilon(1-2\eta)})$ for some fixed constant c.*

Learning of sparse parities with noise is a well-studied open problem in learning theory closely related to problems in coding theory and cryptography. It is known to be at least as hard as learning of DNF expression and juntas over the uniform distribution (Feldman et al., 2009). The trivial algorithm for learning parities on k variables from random examples corrupted by random noise of rate η takes time $n^k \cdot \mathrm{poly}(\frac{1}{1-2\eta})$. The only known improvement to this is an elegant algorithm of Valiant (2012) which runs in time $n^{0.8k} \cdot \mathrm{poly}(\frac{1}{1-2\eta})$.

These results suggest that agnostic learning of monotone submodular functions in time $n^{o(\epsilon^{-2/3})}$ would require a breakthrough in our understanding of these long-standing open problems. In particular, a running time such as $2^{\mathrm{poly}(1/\epsilon)}\mathrm{poly}(n)$, which we achieve in the PAC model, cannot be achieved for agnostic learning of submodular functions. In other words, we show that the agnostic learning algorithm of Cheraghchi et al. (2012) is likely close to optimal. We note that this lower bound does not hold for boolean submodular functions. Monotone boolean submodular functions are disjunctions and hence are agnostically learnable in $n^{O(\log(1/\epsilon))}$ time. For further details on lower bounds we refer the reader to Section 6.

1.2 Related Work

Below we briefly mention some of the other related work. We direct the reader to (Balcan and Harvey, 2012) for a detailed survey. Balcan and Harvey study learning of submodular functions without assumptions on the distribution and also require that the algorithm output a value which is within a multiplicative approximation factor of the true value with probability $\geq 1 - \epsilon$ (the model is referred to as *PMAC learning*). This is a very demanding setting and indeed one of the main results in (Balcan and Harvey, 2012) is a factor-$\sqrt[3]{n}$ inapproximability bound for submodular functions. This notion of approximation is also considered in subsequent works (Badanidiyuru et al., 2012, Balcan et al., 2012) where upper and lower approximation bounds are given for other related classes of functions such as XOS and subadditive. The lower bound of Balcan and Harvey (2012) also implies hardness of learning of submodular function with ℓ_1 (or ℓ_2) error: it is impossible to learn a submodular function $f : \{0,1\}^n \to [0,1]$ in $\mathrm{poly}(n)$ time within any nontrivial ℓ_1 error over general distributions. We emphasize that these strong lower bounds rely on a very specific distribution concentrated on a sparse set of points, and show that this setting is very different from the setting of uniform/product distributions which is the focus of this paper.

For product distributions, Balcan and Harvey show that 1-Lipschitz submodular functions of minimum nonzero value at least 1 have concentration properties implying a PMAC algorithm providing an $O(\log \frac{1}{\epsilon})$-factor approximation except for an ϵ-fraction of points, using $O(\frac{1}{\epsilon} n \log n)$ samples (Balcan and Harvey, 2012). In our setting, we have no assumption on the minimum nonzero value, and we are interested in the additive ℓ_1-error rather than multiplicative approximation.

Gupta et al. (2011) show that submodular functions can be ϵ-approximated by a collection of $n^{O(1/\epsilon^2)}$ ϵ^2-Lipschitz submodular functions. Each ϵ^2-Lipschitz submodular function can be ϵ-approximated by a constant. This leads to a learning algorithm running in time $n^{O(1/\epsilon^2)}$, which however requires value oracle access to the target function, in order to build the collection. Their decomposition is also the basis of our approach. We remark that our algorithm can be directly translated into a faster algorithm for the private data release which motivated the problem in (Gupta et al., 2011). However, for one of their main examples which is privately releasing disjunctions one does not need the full generality of submodular functions. Coverage functions suffice and for those even faster algorithms are now known (Cheraghchi et al., 2012, Feldman and Kothari, 2013).

In a concurrent work, Feldman and Kothari (2013) consider learning of coverage functions. Coverage functions are a simple subclass of submodular functions which can be characterized as non-negative combinations of monotone disjunctions. They show that over the uniform distribution any coverage function can be approximated by a polynomial of degree $\log(1/\epsilon)$ over $O(1/\epsilon^2)$ variables and also prove that coverage functions can be PAC learned in fully-polynomial time (that is, with polynomial dependence on both n and $1/\epsilon$). Note that our lower bounds rule out the possibility of such algorithms for all submodular functions. Their techniques are different from ours (aside from applications of standard Fourier representation-based algorithms).

2 Preliminaries

We work with Boolean functions on $\{0,1\}^n$. Let $\mathcal{U}$ denote the uniform distribution over $\{0,1\}^n$.

Submodularity A set function $f : 2^N \to \mathbb{R}$ is submodular if $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$ for all $A, B \subseteq N$. In this paper, we work with an equivalent description of set functions as functions on the hypercube $\{0,1\}^n$.

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 $\partial_i f(x) = f(x_{i\leftarrow 1}) - f(x_{i\leftarrow 0})$. A function $f : \{0,1\}^n \to \mathbb{R}$ is submodular iff $\partial_i f$ is a non-increasing function for each $i \in [n]$, or equivalently, for all $i \neq j$, $\partial_{i,j} f(x) = \partial_i(\partial_j f(x)) \leq 0$. A function $f : \{0,1\}^n \to \mathbb{R}$ is α-**Lipschitz** if $\partial_i f(x) \in [-\alpha, \alpha]$ for all $i \in [n], x \in \{0,1\}^n$.

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 $\epsilon\|f\|_1$ (Cheraghchi et al., 2012). In fact, it is known that for nonnegative submodular functions, $\|f\|_1 = \mathbf{E}[f] \geq \frac{1}{4}\|f\|_\infty$ and hence this does not make much difference. If we scale $f(x)$ by $1/(4\|f\|_1)$, we obtain a function with values in $[0,1]$. Learning this function within an additive error of ϵ is equivalent to learning the original function within an error of $4\epsilon\|f\|_1$.

Decision Trees: We use $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ to refer to n functions on $\{0,1\}^n$ such that $\mathbf{x}_i(x) = x_i$. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. We represent real-valued functions over $\{0,1\}^n$ using binary decision trees in which each leaf can itself be any real-valued function. Specifically, a function is represented as binary tree T in which each internal node labeled by some variable $\mathbf{x} \in X$ and each leaf ℓ labeled by some real-valued function f_ℓ over variables not restricted on the path to the leaf. We refer to a decision tree in which each leaf is labeled by a function from some set of functions $\mathcal{F}$ as $\mathcal{F}$-valued. If $\mathcal{F}$ contains only constants from the domain of the function then we obtain the usual decision trees.

For a decision tree T with variable $\mathbf{x}_r \in X$ at the root we denote by T_0 (T_1) the left subtree of T (the right subtree, respectively). The value of the tree on a point x is computed in the standard way: if the tree is a leaf ℓ then $T(x) = f_\ell(x_{X[v]})$, where $X[v]$ is the set of indices of variables which are not restricted on the path to ℓ and $x_{X[v]}$ is the substring of x containing all the coordinates in $X[v]$. If T is not a leaf then $T(x) = T_{\mathbf{x}_r(x)}(x)$ where $\mathbf{x}_r$ is the variable at the root of T.

The *rank* of a decision tree T is defined as follows (Ehrenfeucht and Haussler, 1989). If T is a leaf, then $\mathtt{rank}(T) = 0$. Otherwise:

$$\mathtt{rank}(T) = \begin{cases} \max\{\mathtt{rank}(T_0), \mathtt{rank}(T_1)\} & \text{if } \mathtt{rank}(T_0) \neq \mathtt{rank}(T_1); \\ \mathtt{rank}(T_0) + 1, & \text{otherwise.} \end{cases}$$

The *depth* of a node v in a tree T is the length of the path the root of T to v. The depth of a tree is the depth of its deepest leaf. For any node $v \in T$ we denote by $T[v]$ the sub-tree rooted at that node. We also use T to refer to the function computed by T.

Fourier Analysis on the Boolean Cube We define the notions of inner product and norms, which we consider with respect to $\mathcal{U}$. For two functions $f, g : \{0,1\}^n \to \mathbb{R}$, the inner product of f and g is defined as $\langle f, g\rangle = \mathbf{E}_{x\sim\mathcal{U}}[f(x) \cdot g(x)]$. The ℓ_1 and ℓ_2 norms of f 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.

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 parities form an orthonormal basis for functions on $\{0,1\}^n$ under the inner product product with respect to the uniform distribution. Thus, every function $f : \{0,1\}^n \to \mathbb{R}$ can be written as a real linear combination of parities. The coefficients of the linear combination are referred to as Fourier coefficients of f. For $f : \{0,1\}^n \to \mathbb{R}$ and $S \subseteq [n]$, the Fourier coefficient $\hat{f}(S)$ is given by $\hat{f}(S) = \langle f, \chi_S\rangle$. For any Fourier coefficient $\hat{f}(S)$, $|S|$ is called the *degree* of the coefficient.

The Fourier expansion of f is given by $f(x) = \sum_{S\subseteq[n]} \hat{f}(S)\chi_S(x)$. The degree of highest degree non-zero Fourier coefficient of f is referred to as the *Fourier degree* of f. Note that Fourier degree of f is exactly the polynomial degree of f when viewed over $\{-1,1\}^n$ instead of $\{0,1\}^n$ and therefore it is also equal to the polynomial degree of f over $\{0,1\}^n$. Let $f : \{0,1\}^n \to \mathbb{R}$ and $\hat{f} : 2^{[n]} \to \mathbb{R}$ be its Fourier Transform.

The *spectral ℓ_1 norm* of f is defined as

$$||\hat{f}||_1 = \sum_{S \subseteq [n]} |\hat{f}(S)|.$$

The Fourier transform of partial derivatives satisfies: $\partial_i f(x) = 2 \sum_{S \ni i} \hat{f}(S) \chi_{S \setminus \{i\}}(x)$, and $\partial_{i,j} f(x) = 4 \sum_{S \ni i,j} \hat{f}(S) \chi_{S \setminus \{i,j\}}(x)$.

Learning Models Our learning algorithms are in one of two standard models of learning. The first one assumes that the learner has access to random examples of an unknown function from a known set of functions. This model is a generalization of Valiant's PAC learning model to real-valued functions (Valiant, 1984, Haussler, 1992).

Definition 2.1 (PAC ℓ_1-learning)**.** *Let $\mathcal{F}$ be a class of real-valued functions on $\{0,1\}^n$ and let $\mathcal{D}$ be a distribution on $\{0,1\}^n$. An algorithm $\mathcal{A}$ PAC learns $\mathcal{F}$ on $\mathcal{D}$, if for every $\epsilon > 0$ and any target function $f \in \mathcal{F}$, given access to random independent samples from $\mathcal{D}$ labeled by f, with probability at least $\frac{2}{3}$, $\mathcal{A}$ returns a hypothesis h such that $\mathbf{E}_{x \sim \mathcal{D}}[|f(x) - h(x)|] \leq \epsilon$. $\mathcal{A}$ is said to be* proper if $h \in \mathcal{F}$.

While in general Valiant's model does not make assumptions on the distribution $\mathcal{D}$, here we only consider the *distribution-specific* version of the model in which the distribution is fixed and is uniform over $\{0,1\}^n$. The error parameter ϵ in the Boolean case measures probability of misclassification. 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 (Haussler, 1992, Kearns et al., 1994) (for example as a result of noise in the labels).

Definition 2.2 (Agnostic ℓ_1-learning)**.** *Let $\mathcal{F}$ be a class of real-valued functions from $\{0,1\}^n$ to $[0,1]$ and let $\mathcal{D}$ be any fixed distribution on $\{0,1\}^n$. For any function f, let opt$(f, \mathcal{F})$ be defined as:*

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

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

$$\mathbf{E}_{x \sim \mathcal{D}}[|h(x) - f(x)|] \leq \textit{opt}(f, \mathcal{F}) + \epsilon.$$

The ℓ_2 versions of these models are defined analogously.

3 Approximation of Submodular Functions by Low-Rank Decision Trees

We now prove that any bounded submodular function can be represented as a low-rank decision tree with α-Lipschitz submodular functions in the leaves. Our construction follows closely the construction of Gupta et al. (2011). They show that for every submodular f there exists a decomposition of $\{0,1\}^n$ into $n^{O(1/\alpha)}$ disjoint regions restricted to each of which f is α-Lipschitz submodular. In essence, we give a binary decision tree representation of the decomposition from (Gupta et al., 2011) and then prove that the decision tree has rank $O(1/\alpha)$.

Theorem 3.1. *Let $f : \{0,1\}^n \to [0,1]$ be a submodular function and $\alpha > 0$. Let $\mathcal{F}_\alpha$ denote the set of all α-Lipschitz submodular functions with range $[0,1]$ over at most n Boolean variables. Then f can be computed by an $\mathcal{F}_\alpha$-valued binary decision tree T of rank $r \leq 2/\alpha$.*

We first prove the claim that decomposes a submodular function f into regions where f where discrete derivatives of f are upper-bounded by α everywhere: we call this property α-monotone decreasing.

Definition 3.2. *For $\alpha \in \mathbb{R}$, f is α-monotone decreasing if for all $i \in [n]$ and $x \in \{0,1\}^n$, $\partial_i f(x) \leq \alpha$.*

We remark that α-Lipschitzness is equivalent to discrete derivatives being in the range $[-\alpha, \alpha]$, i.e. f as well as $-f$ being α-monotone decreasing.

Lemma 3.3. *For $\alpha > 0$ let $f : \{0,1\}^n \to [0,1]$ be a submodular function. Let $\mathcal{M}_\alpha$ denote the set of all α-monotone decreasing submodular functions with range $[0,1]$ over at most n Boolean variables. f can be computed by a $\mathcal{M}_\alpha$-valued binary decision tree T of rank $r \leq 1/\alpha$.*

Proof. The tree T is constructed recursively as follows: if $n = 0$ then the function is a constant which can be computed by a single leaf. If f is α-monotone decreasing then T is equal to the leaf computing f. Otherwise, if f is not α-monotone decreasing then there exists $i \in [n]$ and $z \in \{0,1\}^n$ such that $\partial_i f(z) > \alpha$. In fact, submodularity of f implies that $\partial_i f$ is monotone decreasing and, in particular, $\partial_i f(\bar{0}) \geq \partial_i f(z) > \alpha$. We label the root with $\mathbf{x}_i$ and build the trees T_0 and T_1 for f restricted to points x such that $x_i = 0$ and $x_i = 1$, respectively (viewed as a function over $\{0,1\}^{n-1}$). Note that both restrictions preserve submodularity and α-monotonicity of f.

By definition, this binary tree computes $f(x)$ and its leaves are α-monotone decreasing submodular functions. It remains to compute the rank of T. For any node $v \in T$, we let $X[v] \subseteq [n]$ be the set of indices of variables that are not set on the path to v, let $\bar{X}[v] = [n] \setminus X[v]$ and let $y[v] \in \{0,1\}^{\bar{X}[v]}$ denote the values of the variables that were set. Let $\{0,1\}^{X[v]}$ be the subcube of points in $\{0,1\}^n$ that reach v, namely points x such that $x_{X[v]} = y[v]$. Let $f[v](x) = T[v](x)$ be the restriction of f to the subcube. Note that the vector of all 0's, $\bar{0}$ in the $\{0,1\}^{X[v]}$ subcube corresponds to the point which equals $y[v]$ on coordinates in $\bar{X}[v]$ and 0 on all other coordinates. We refer to this point as $x[v]$.

Let $M = \max_x\{f(x)\}$. We prove by induction on the depth of $T[v]$ that for any node $v \in T$,

$$\texttt{rank}(T[v]) \leq \frac{M - f[v](\bar{0})}{\alpha}. \tag{1}$$

This is obviously true if v is a leaf. Now, let v be an internal node v with label $\mathbf{x}_i$. Let v_0 and v_1 denote the roots of $T[v]_0$ and $T[v]_1$, respectively. For v_0, $x[v_0] = x[v]$ and therefore $f[v](\bar{0}) = f[v_0](\bar{0})$. By inductive hypothesis, this implies that

$$\texttt{rank}[T[v_0]] \leq \frac{M - f[v_0](\bar{0})}{\alpha} = \frac{M - f[v](\bar{0})}{\alpha}\,. \tag{2}$$

We know that $\partial_i f[v](\bar{0}) > \alpha$. By definition, $\partial_i f[v](\bar{0}) = f[v](\bar{0}_{i\leftarrow 1}) - f[v](\bar{0})$. At the same time, $f[v](\bar{0}_{i\leftarrow 1}) = f(x[v]_{i\leftarrow 1}) = f(x[v_1]) = f[v_1](\bar{0})$. Therefore, $f[v_1](\bar{0}) \geq f[v](\bar{0}) + \alpha$. By the inductive hypothesis, this implies that

$$\texttt{rank}[T[v_1]] \leq \frac{M - f[v_1](\bar{0})}{\alpha} \leq \frac{M - f[v](\bar{0}) - \alpha}{\alpha} = \frac{M - f[v](\bar{0})}{\alpha} - 1\,. \tag{3}$$

Combining equations (2) and (3) and using the definition of the rank we obtain that equation (1) holds for v.

The claim now follows since f has range $[0,1]$ and thus $M \leq 1$ and $f(\bar{0}) \geq 0$. □

We note that for monotone functions Lemma 3.3 implies Theorem 3.1 since discrete derivatives of a monotone function are non-negative. As in the construction in (Gupta et al., 2011), the extension to the non-monotone case is based on observing that for any submodular function f, the function $\bar{f}(x) = f(\neg x)$ is also submodular, where $\neg x$ is obtained from x by flipping every bit.

Proof of Theorem 3.1. We first apply Lemma 3.3 to obtain an $\mathcal{M}_\alpha$-valued decision tree T' for f of rank $\leq 1/\alpha$. Now let ℓ be any leaf of T' and let $f[\ell]$ denote f restricted to ℓ. As before, let $X[\ell] \subseteq [n]$ be the set of indices of variables that are not restricted on the path to ℓ and let $\{0,1\}^{X[\ell]}$ be the subcube of points in $\{0,1\}^n$ that reach ℓ. We now use Lemma 3.3 to obtain an $\mathcal{M}_\alpha$-valued decision tree T_ℓ for $\overline{f[\ell]}$ of rank $\leq 1/\alpha$. We denote by $\neg T_\ell$ the tree computing the function $T_\ell(\neg z)$. It is obtained from T_ℓ by swapping the subtrees of each node and replacing each function $g(z)$ in a leaf with $g(\neg z)$. We replace each leaf ℓ of T' by $\neg T_\ell$ and let T be the resulting tree. To prove the theorem we establish the following properties of T.

1. Correctness: we claim that $T(x)$ computes $f(x)$. To see this note that for each leaf ℓ of T', $\neg T_\ell(z)$ computes $T_\ell(\neg z) = \overline{f[\ell]}(\neg z) = f[\ell](z)$. Hence $T(x) = T'(x) = f(x)$.

2. α-Lipschitzness of leaves: by our assumption, $f[\ell]$ is an α-monotone decreasing function over $\{0,1\}^{X[\ell]}$ and therefore $\partial_i f[\ell](z) \geq -\alpha$ for all $i \in X[\ell]$ and $z \in \{0,1\}^{X[\ell]}$. This means that for all $i \in X[\ell]$ and $z \in \{0,1\}^{X[\ell]}$,

$$\partial_i \overline{f[\ell]}(z) = -\partial_i f[\ell](\neg z) \leq \alpha. \tag{4}$$

   Further, let κ be a leaf of T_ℓ computing a function $\overline{f[\ell]}[\kappa]$. By Lemma 3.3, $\overline{f[\ell]}[\kappa]$ is α-monotone decreasing. Together with equation 4 this implies that $\overline{f[\ell]}[\kappa]$ is α-Lipschitz. In $\neg T_\ell$, $\overline{f[\ell]}[\kappa](z)$ is replaced by $\overline{f[\ell]}[\kappa](\neg z)$. This operation preserves α-Lipschitzness and therefore all leaves of T are α-Lipschitz functions.

3. Submodularity of the leaf functions: for each leaf ℓ, $f[\ell]$ is submodular simply because it is a restriction of f to a subcube.

4. Rank: by Lemma 3.3, $\texttt{rank}(T') \leq 2/\alpha$ and for every leaf ℓ of T', $\texttt{rank}(\neg T_\ell) = \texttt{rank}(T_\ell) \leq 1/\alpha$. As can be easily seen from the definition of rank, replacing each leaf of T' by a tree of rank at most $1/\alpha$ can increase the rank of the resulting tree by at most $1/\alpha$. Hence the rank of T is at most $2/\alpha$.

□

3.1 Approximation of Leaves

An important property of the decision tree representation is that it decomposes a function into disjoint regions. This implies that approximating the function over the whole domain can be reduced to approximating the function over individual regions with the same error parameter. Then, as in (Gupta et al., 2011), we can use concentration properties of α-Lipschitz submodular functions on the uniform distribution $\mathcal{U}$ over $\{0,1\}^n$ to approximate each α-Lipschitz submodular functions by a constant.

Formally we state the following lemma which allows the use of any loss function L.

Lemma 3.4. *For a set of functions $\mathcal{F}$, let T be an $\mathcal{F}$-valued binary decision tree, D be any distribution over $\{0,1\}^n$ and $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be any real-valued (loss) function. For each leaf $\ell \in T$, let $D[\ell]$ be the distribution over $\{0,1\}^{X[\ell]}$ that equals D conditioned on x reaching ℓ; let g_ℓ be a function that satisfies*

$$\mathbf{E}_{z \sim D[\ell]} \left[L\left(T[\ell](z), g_\ell(z)\right) \right] \leq \epsilon.$$

Let T' be the tree obtained from T by replacing each function in a leaf ℓ with the corresponding g_ℓ. Then $\mathbf{E}_{x \sim D}[L(T(x), T'(x))] \leq \epsilon$.

Proof. For a leaf $\ell \in T$, let $y[\ell] \in \{0,1\}^{\bar{X}[\ell]}$ denote the values of the variables that were set on the path to ℓ. Note that the subcube $\{0,1\}^{X[\ell]}$ corresponds to the points $x \in \{0,1\}^n$ such that $x_{X[\ell]} = y[\ell]$.

$$\begin{aligned}
\mathbf{E}_{x\sim D}[L(T(x),T'(x))] &= \sum_{\ell\in T} \mathbf{E}_{x\sim D}\left[L(T(x),T'(x)) \mid x_{X[\ell]} = y[\ell]\right] \cdot \Pr_{x\sim D}\left[x_{X[\ell]} = y[\ell]\right] \\
&= \sum_{\ell\in T} \mathbf{E}_{z\sim D[\ell]}\left[L(T[\ell](z), g_\ell(z))\right] \cdot \Pr_{x\sim D}\left[x_{X[\ell]} = y[\ell]\right] \\
&\leq \sum_{\ell\in T} \epsilon \cdot \Pr_{x\sim D}\left[x_{X[\ell]} = y[\ell]\right] = \epsilon\,.
\end{aligned}$$

□

It is known that 1-Lipschitz submodular functions satisfy strong concentration properties over the uniform distribution $\mathcal{U}$ over $\{0,1\}^n$ (Boucheron et al., 2000, Vondrák, 2010, Balcan and Harvey, 2012), with standard deviation $O(\sqrt{\mathbf{E}[f]})$ and exponentially decaying tails. For our purposes we do not need the exponential tail bounds and instead we state the following simple bound on variance.

Lemma 3.5. *For any α-Lipschitz submodular function $f : \{0,1\}^n \to \mathbb{R}_+$,*

$$\mathbf{Var}_{\mathcal{U}}[f] \leq 2\alpha \cdot \mathbf{E}_{\mathcal{U}}[f].$$

Proof. By the Efron-Stein inequality (see (Boucheron et al., 2000)),

$$\mathbf{Var}_{\mathcal{U}}[f] \leq \frac{1}{2}\sum_{i\in[n]} \mathbf{E}_{\mathcal{U}}[(\partial_i f)^2] \leq \frac{1}{2}\max_{i\in[n]} \mathbf{E}_{\mathcal{U}}[|\partial_i f|] \cdot \sum_{i\in[n]} \mathbf{E}_{\mathcal{U}}[|\partial_i f|] \leq \alpha \cdot \frac{1}{2}\sum_{i\in[n]} \mathbf{E}_{\mathcal{U}}[|\partial_i f|]\,.$$

We can now use the fact that non-negative submodular functions are 2-self-bounding (Vondrák, 2010), and hence $\sum_{i\in[n]} \mathbf{E}_{\mathcal{U}}[|\partial_i f|] = 2\mathbf{E}_{x\sim\mathcal{U}}[\sum_{i:f(x\oplus e_i)<f(x)}(f(x) - f(x\oplus e_i))] \leq 4\mathbf{E}_{\mathcal{U}}[f]$. □

We can now finish the proof of Theorem 1.1.

Proof of Theorem 1.1. Let T' be the $\mathcal{F}_\alpha$-valued decision tree for f given by Theorem 3.1 with $\alpha = \epsilon^2/2$. For every leaf ℓ we replace the function $T'[\ell]$ at that leaf by the constant $\mathbf{E}_{\mathcal{U}}[T'[\ell]]$ (here the uniform distribution is over $\{0,1\}^{X[\ell]}$) and let T be the resulting tree.

Cor. 3.5 implies that for any $\epsilon^2/2$-Lipschitz submodular function $g : \{0,1\}^m \to [0,1]$, $\mathbf{Var}_{\mathcal{U}}[g] = \mathbf{E}_{\mathcal{U}}[(g - \mathbf{E}_{\mathcal{U}}[g])^2] \leq 2\frac{\epsilon^2}{2}\mathbf{E}_{\mathcal{U}}[g] \leq \epsilon^2$. For every leaf $\ell \in T'$, $T'[\ell]$ is $\epsilon^2/2$-Lipschitz and hence,

$$\mathbf{E}_{\mathcal{U}}[(T'[\ell](z) - T[\ell](z))^2] = \mathbf{E}_{\mathcal{U}}[(T'[\ell](z) - \mathbf{E}_{\mathcal{U}}[T'[\ell]])^2] \leq \epsilon^2\,.$$

By Lemma 3.4 (with $L(a,b) = (a-b)^2$), we obtain that $\mathbf{E}_{\mathcal{U}}[(T(x) - f(x))^2] \leq \epsilon^2$. □

4 Approximation of Low-Rank Decision Trees by Shallow Decision Trees

We show that over any constant-bounded product distribution D, a decision tree of rank r can be ϵ-approximated by a decision tree of depth $O(r + \log(1/\epsilon))$. The approximating decision tree is simply the original tree pruned at depth $d = O(r + \log(1/\epsilon))$.

For a vector $\mu \in [0,1]^n$ we denote by D_μ the product distribution over $\{0,1\}^n$, such that $\Pr_{D_\mu}[x_i = 1] = \mu_i$. For $\alpha \in [0, 1/2]$ a product distribution D_μ is α-bounded if $\mu \in [\alpha, 1-\alpha]^n$. For a decision tree T and integer $d \geq 0$ we denote by $T^{\leq d}$ a decision tree in which all internal nodes at depth d are replaced by a leaf computing constant 0.

Theorem 4.1. *(Theorem 1.2 restated) For a set of functions $\mathcal{F}$ let T be a $\mathcal{F}$-valued decision tree of rank r, and let D_μ be an α-bounded product distribution for some $\alpha \in (0, 1/2]$. Then for any integer $d \geq 0$,*

$$\Pr_{D_\mu}[T^{\leq d}(x) \neq T(x)] \leq 2^{r-1} \cdot \left(1 - \frac{\alpha}{2}\right)^d .$$

In particular, for $d = \lfloor (r + \log(1/\epsilon))/\log(2/(2-\alpha)) \rfloor$ we get that $\Pr_{D_\mu}[T^{\leq d}(x) \neq T(x)] \leq \epsilon$.

Proof. Our proof is by induction on the pruning depth d. If T is a leaf, the statement trivial since $T^{\leq d}(x) \equiv T(x)$ for any $d \geq 0$. For $d = 0$ and $r \geq 1$, $2^{r-1} \cdot \left(1 - \frac{\alpha}{2}\right)^0 \geq 1$. We now assume that the claim is true for all pruning depths $0, \ldots, d-1$.

At least one of the subtrees T_0 and T_1 has rank $r-1$. Assume, without loss of generality that this is T_0. Let $\mathbf{x}_i$ be the label of the root node of T.

$$\Pr_{D_\mu}[T^{\leq d}(x) \neq T(x)] = (1 - \mu_i) \Pr_{D_\mu}[T_0^{\leq d-1}(x) \neq T_0(x)] + \mu_i \cdot \Pr_{D_\mu}[T_1^{\leq d-1}(x) \neq T_1(x)] .$$

By our inductive hypothesis,

$$\Pr_{D_\mu}[T_0^{\leq d-1}(x) \neq T_0(x)] \leq 2^{r-2} \cdot \left(1 - \frac{\alpha}{2}\right)^{d-1}$$

and

$$\Pr_{D_\mu}[T_0^{\leq d-1}(x) \neq T_0(x)] \leq 2^{r-1} \cdot \left(1 - \frac{\alpha}{2}\right)^{d-1} .$$

Combining these we get that

$$\begin{aligned}
\Pr_{D_\mu}[T^{\leq d}(x) \neq T(x)] &\leq (1 - \mu_i) 2^{r-2} \cdot \left(1 - \frac{\alpha}{2}\right)^{d-1} + \mu_i \cdot 2^{r-1} \cdot \left(1 - \frac{\alpha}{2}\right)^{d-1} \\
&\leq \alpha \cdot 2^{r-2} \cdot \left(1 - \frac{\alpha}{2}\right)^{d-1} + (1 - \alpha) \cdot 2^{r-1} \cdot \left(1 - \frac{\alpha}{2}\right)^{d-1} \\
&= \frac{1}{1 - \frac{\alpha}{2}} \left(\frac{\alpha}{2} + (1 - \alpha)\right) 2^{r-1} \cdot \left(1 - \frac{\alpha}{2}\right)^d = 2^{r-1} \cdot \left(1 - \frac{\alpha}{2}\right)^d .
\end{aligned}$$

□

For the uniform distribution we get error of at most ϵ for $d = (r + \log(1/\epsilon))/\log(4/3) < \frac{5}{2}(r + \log(1/\epsilon))$.

An immediate corollary of Theorems 4.1 and 1.1 is that every submodular function can be ϵ-approximated over the uniform distribution by a binary decision tree of depth $O(1/\epsilon^2)$ (Corollary 1.3).

Kushilevitz and Mansour (1993) showed that the spectral ℓ_1 norm of a decision tree of size s is at most s. Therefore we can immediately conclude that:

Corollary 4.2. *Let $f : \{0,1\}^n \to [0,1]$ be a submodular function and $\epsilon > 0$. There exists a function $p : \{0,1\}^n \to [0,1]$ such that $\|p - f\|_2 \leq \epsilon$ and $\|\hat{p}\|_1 = 2^{O(1/\epsilon^2)}$.*

5 Applications

In this section, we give several applications of our structural results to the problem of learning submodular functions.

5.1 PAC Learning

In this section we present our results on learning in the PAC model. We first show how to find $2^{O(1/\epsilon^2)}$ variables that suffice for approximating any submodular function using random examples alone. Using a fairly standard argument we first show that for any function f that is close to a function of low polynomial degree and low spectral ℓ_1 norm (which is satisfied by submodular functions) variables sufficient for approximating f can be found by looking at significant Fourier coefficients of f (the proof is in App. **??**)

Lemma 5.1. *Let $f : \{0,1\}^n \to [0,1]$ be any function such that there exists a function p of Fourier degree d and spectral ℓ_1 norm $\|\hat{p}\|_1 = L$ for which $\|f - p\|_2 \leq \epsilon$. Define*

$$J = \{i \mid \exists S; i \in S, |S| \leq d \text{ and } |\hat{f}(S)| \geq \epsilon^2/L\}.$$

Then $|J| \leq d \cdot L^2/\epsilon^4$ and there exists a function p' of Fourier degree d over variables in J such that $\|f - p\|_2 \leq 2\epsilon$.

Proof. Let

$$\mathcal{S} = \{S \mid |S| \leq d \text{ and } |\hat{f}(S)| \geq \epsilon^2/L\}.$$

By Parseval's identity, there are at most L^2/ϵ^4 sets in $\mathcal{S}$. Clearly, J is the union of all the sets in $\mathcal{S}$. Therefore, the bound on the size of J follows immediately from the fact that each set $S \in \mathcal{S}$ has size at most d.

Let p' be the projection of p to the subspace of $\{\chi_S : S \in \mathcal{S}\}$, that is $p' = \sum_{S \in \mathcal{S}} \hat{p}(S)\chi_S$. Now using Parseval's identity we get that

$$\|f - p\|_2^2 = \sum_{S \subseteq [n]} (\hat{f}(S) - \hat{p}(S))^2 .$$

Now we observe that for any S, $|\hat{f}(S) - \hat{p}(S)| < |\hat{f}(S) - \hat{p'}(S)|$ can happen only when $S \notin \mathcal{S}$ in which case $\hat{p'}(S) = 0$ and $|\hat{f}(S)| \leq \epsilon^2/L$.

$|\hat{p}(S)| \leq 2|\hat{f}(S)|$; hence only when $|\hat{p}(S)| \leq 2\epsilon^2/L$. In this case,

$$(\hat{f}(S) - \hat{p'}(S))^2 - (\hat{f}(S) - \hat{p}(S))^2 = 2\hat{f}(S)\hat{p}(S) - (\hat{p}(S))^2 \leq 2\hat{f}(S)\hat{p}(S) \leq 2|\hat{p}(S)| \cdot \epsilon^2/L .$$

Therefore,

$$\|f - p'\|_2^2 - \|f - p\|_2^2 = \sum_S (\hat{f}(S) - \hat{p'}(S))^2 - (\hat{f}(S) - \hat{p}(S))^2 \leq \frac{2\epsilon^2}{L} \sum_S |\hat{p}(S)| \leq \frac{2\epsilon^2}{L} \cdot \|\hat{p}\|_1 = 2\epsilon^2.$$

This implies that $\|f - p'\|_2^2 \leq 3\epsilon^2$. □

The second and crucial observation that we make is a connection between Fourier coefficient of $\{i, j\}$ of a submodular function and sum of squares of all Fourier coefficients that contain $\{i, j\}$.

Lemma 5.2. *Let $f : \{0,1\}^n \to [0,1]$ be a submodular function and $i, j \in [n]$, $i \neq j$.*

$$|\hat{f}(\{i,j\})| \geq \frac{1}{2} \sum_{S \ni i,j} (\hat{f}(S))^2.$$

Proof.

$$|\hat{f}(\{i,j\})| =^{(a)} \frac{1}{4}|\mathbf{E}_{\mathcal{U}}[\partial_i\partial_j f]| =^{(b)} \frac{1}{4}\mathbf{E}_{\mathcal{U}}[|\partial_i\partial_j f|] \geq^{(c)} \frac{1}{8}\mathbf{E}_{\mathcal{U}}\left[(\partial_i\partial_j f)^2\right] =^{(a)} 2\sum_{S\ni i,j}(\hat{f}(S))^2.$$

Here, (a) follows from the basic properties of the Fourier spectrum of partial derivatives (see Sec. 2); (b) is implied by second partial derivatives of a submodular function being always non-positive; and (c) follows from $|\partial_i\partial_j f|$ having range $[0,2]$ whenever f has range $[0,1]$. □

We can now easily complete the proof of Lemma 1.4.

Proof of Lemma 1.4. The proof relies on two simple observations. The first one is that Lemma 5.1 implies that the set of indices $I_\gamma = \{i \mid \exists S \ni i, |\hat{f}(S)| \geq \gamma\}$ satisfies the conditions of Lemma 1.4 for some $\gamma = 2^{-O(1/\epsilon^2)}$.

Now if $i \in I_\gamma$ then either $|\hat{f}(\{i\})| \geq \gamma$ or, exists $j \neq i$, such that for some $S' \ni i,j$, $|\hat{f}(S')| \geq \gamma$. In the latter case $\sum_{S\ni i,j}(\hat{f}(S))^2 \geq \gamma^2$. By Lemma 5.2 we can conclude that then $|\hat{f}(\{i,j\})| \geq 2\gamma^2$.

This suggests the following simple algorithm for finding J. Estimate degree 1 and 2 Fourier coefficients of f to accuracy $\gamma^2/2$ with confidence at least $5/6$ using random examples (note that $\gamma < 1/2$ and hence degree-1 coefficients are estimated with accuracy at least $\gamma/4$. Let $\tilde{f}(S)$ for $S \subseteq [n]$ of size 1 or 2 denote the obtained estimates. We define

$$J = \left\{i \mid \exists j \in [n], |\tilde{f}(\{i,j\})| \geq 3\gamma^2/2\right\}.$$

If the estimates are correct, then clearly, $I_\gamma \subseteq J$. At the same time, J contains inly indices which belong to a Fourier coefficient of magnitude at least γ^2 and degree at most 2. By Parseval's identity, $|J| \leq 2\|f\|_2^2/\gamma^4 = 2^{O(1/\epsilon^2)}$.

Finally, to bound the running time we observe that, by Chernoff bounds, $O(\log(n)/\gamma^4) = \log(n)\cdot 2^{O(1/\epsilon^2)}$ random examples are sufficient to obtain the desired estimates with confidence of $5/6$. The estimation of the coefficients can be done in $n^2\log(n)\cdot 2^{O(1/\epsilon^2)}$ time. □

Now given a set J that was output by the algorithm in Lemma 1.4 one can simply run the standard low-degree algorithm of Linial et al. (1993) over variables with indices in J to find a linear combination of parities of degree $O(1/\epsilon^2)$, ϵ-close to f. Note that we need to find coefficients of at most $|J|^{O(1/\epsilon^2)} \leq \min\{2^{O(1/\epsilon^4)}, n^{O(1/\epsilon^2)}\}$ parities. This immediately implies Theorem 1.5.

5.2 Agnostic learning with value queries

Our next application is agnostic learning of submodular functions over the uniform distribution with value queries. We give two versions of the agnostic learning algorithm one based on ℓ_1 and the other based on ℓ_2 error. We note that, unlike in the PAC setting where small ℓ_2 error also implied small ℓ_1 error, these two versions are incomparable and are also based on different algorithmic techniques. The agnostic learning techniques we use are not new but we give attribute-efficient versions of those techniques using an attribute-efficient agnostic learning of parities from (Feldman, 2007).

For the ℓ_2 agnostic learning algorithm we need a known observation (e.g. (Gopalan et al., 2008)) that the algorithm of Kushilevitz and Mansour (1993) can be used to obtain agnostic learning relative to ℓ_2-norm of all functions with spectral ℓ_1 norm of L in time $\mathrm{poly}(n, L, 1/\epsilon)$ (we include a proof in App. A). We also observe that in order to learn agnostically decision trees of depth d it is sufficient to restrict the attention to

significant Fourier coefficients of degree at most d. We can exploit this observation to improve the number of value queries used for learning by using the attribute-efficient agnostic parity learning from (Feldman, 2007) in place of the KM algorithm. Specifically, we first prove the following attribute-efficient version of agnostic learning of functions with low spectral ℓ_1-norm (the proof appears in App. A).

Theorem 5.3. *For $L > 0$, we define $\mathcal{C}_L^d$ as $\{p(x) \mid \|\hat{p}\|_1 \leq L$ and $\mathtt{degree}(p) \leq d\}$. 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 $\|f - h\|_2 \leq \Delta + \epsilon$, where $\Delta = \min_{p \in \mathcal{C}_L}\{\|f - p\|_2\}$. Further, $\mathcal{A}$ runs in time $\mathrm{poly}(n, L, 1/\epsilon)$ and uses $\mathrm{poly}(d, \log(n), L, 1/\epsilon)$ value queries.*

Together with Cor. 4.2 this implies Theorem 1.6.

Gopalan et al. (2008) give the ℓ_1 version of agnostic learning for functions of low spectral ℓ_1 norm. Together with Cor. 4.2 this implies an ℓ_1 agnostic learning algorithm for submodular functions using $\mathrm{poly}(n, 2^{1/\epsilon^2})$ time and queries. There is no known attribute-efficient version of the algorithm of Gopalan et al. (2008) and their analysis is relatively involved. Instead we use our approximate representation by decision trees to invoke a substantially simpler algorithm for agnostic learning of decision trees based on agnostic boosting (Kalai and Kanade, 2009, Feldman, 2010). In this algorithm it is easy to use attribute-efficient agnostic learning of parities (Feldman, 2007) (restated in Th. A.1) to reduce the query complexity of the algorithm. Formally we give the following attribute-efficient algorithm for learning $[0,1]$-valued decision trees.

Theorem 5.4. *Let $\mathtt{DT}_{[0,1]}(r)$ denote the class of all $[0,1]$-valued decision trees of rank-r on $\{0,1\}^n$. There exists an algorithm $\mathcal{A}$ that given $\epsilon > 0$ and access to value queries of any $f : \{0,1\}^n \to \{0,1\}$, with probability at least $2/3$, outputs a function $h : \{0,1\}^n \to [0,1]$, such that $\|f - h\|_1 \leq \Delta + \epsilon$, where $\Delta = \min_{g \in \mathtt{DT}_{[0,1]}(r)}\{\|f - g\|_1\}$. Further, $\mathcal{A}$ runs in time $\mathrm{poly}(n, 2^r, 1/\epsilon)$ and uses $\mathrm{poly}(\log n, 2^r, 1/\epsilon)$ value queries.*

Combining Theorems 5.4 and 1.1 gives the following agnostic learning algorithm for submodular functions (the proof is in App. A).

Theorem 5.5. *Let $\mathcal{C}_s$ denote the class of all submodular functions from $\{0,1\}^n$ to $[0,1]$. There exists an algorithm $\mathcal{A}$ that given $\epsilon > 0$ and access to value queries of any real-valued f, with probability at least $2/3$, outputs a function h, such that $\|f - h\|_1 \leq \Delta + \epsilon$, where $\Delta = \min_{g \in \mathcal{C}_s}\{\|f - g\|_1\}$. Further, $\mathcal{A}$ runs in time $\mathrm{poly}(n, 2^{1/\epsilon^2})$ and using $\mathrm{poly}(\log n, 2^{1/\epsilon^2})$ value queries.*

6 Lower Bounds

6.1 Computational Lower Bounds for Agnostic Learning of Submodular Functions

In this section we show that the existence of an algorithm for agnostically learning even *monotone and symmetric*[1] *submodular functions* (i.e. concave functions of $\sum x_i$) to an accuracy of any $\epsilon > 0$ in time $n^{o(1/\epsilon^{2/3})}$ would yield a faster algorithm for *learning sparse parities with noise* (SLPN from now) which is a well known and notoriously hard problem in computational learning theory.

We begin by stating the problems of Learning Parities with Noise (LPN) and its variant, learning sparse parities with noise (SLPN). We say that random examples of a function f have noise of rate η if the label of a random example equals $f(x)$ with probability $1 - \eta$ and $-f(x)$ with probability η.

[1] In this context, we call a function $f : \{0,1\}^n \to \mathbb{R}$ symmetric if $f(x)$ depends only on $\sum x_i$. This is different from the notion of a symmetric set function, which usually means the condition $f(S) = f(\bar{S})$.

Problem 6.1 (Learning Parities with Noise)**.** *For $\eta \in (0, 1/2)$, the problem of learning parities with noise η is the problem of finding (with probability at least $2/3$) the set $S \subseteq [n]$, given access to random examples with noise of rate η of parity function χ_S. For $k \leq n$ the learning of k-sparse parities with noise η is the same problem with an additional condition that $|S| \leq k$.*

The best known algorithm for the LPN problem with constant noise rate is by Blum et al. (2003) and runs in time $2^{O(n/\log n)}$. The fastest known algorithm for learning k-sparse parities with noise η is a recent breakthrough result of Valiant (2012) which runs in time $O(n^{0.8k}\mathrm{poly}(\frac{1}{1-2\eta}))$.

Kalai et al. (2008) and Feldman (2012) prove hardness of agnostic learning of majorities and conjunctions, respectively, based on correlation of concepts in these classes with parities. In both works it is implicit that if for every set $S \subseteq [n]$, a concept class $\mathcal{C}$ contains a function f_S that has significant correlation with χ_S (or $\widehat{f_S}(S)$) then learning of parities with noise can be reduced to agnostic learning of $\mathcal{C}$. We now present this reduction in a general form.

Lemma 6.2. *Let $\mathcal{C}$ be a class of functions mapping $\{0,1\}^n$ into $[-1,1]$. Suppose, there exist $\gamma > 0$ and $k \in \mathbb{N}$ such that for every $S \subseteq [n]$, $|S| \leq k$, there exists a function, $f_S \in \mathcal{C}$, such that $|\widehat{f_S}(S)| \geq \gamma$. If there exists an algorithm $\mathcal{A}$ that learns the class $\mathcal{C}$ agnostically to accuracy ϵ in time $T(n, \frac{1}{\epsilon})$ then, there exists an algorithm $\mathcal{A}'$ that learns k-sparse parities with noise $\eta \leq 1/2$ in time $\mathrm{poly}(n, \frac{1}{(1-2\eta)\gamma}) + 2T(n, \frac{2}{(1-2\eta)\gamma})$.*

Proof. Let χ_S be the target parity with $|S| \leq k$. We run algorithm $\mathcal{A}'$ with $\epsilon = (1-2\eta)\gamma/2$ on the noisy examples and let h be the hypothesis it outputs. We also run algorithm $\mathcal{A}'$ with $\epsilon = (1-2\eta)\gamma/2$ on the negated noisy examples and let h' be the hypothesis it outputs.

Now let $f_S \in \mathcal{C}$ be the function such that $|\widehat{f_S}(S)| \geq \gamma$. Assume without loss of generality that $\widehat{f_S}(S) \geq \gamma$ (otherwise we will use the same argument on the negation of f_S). Let $\mathcal{N}^\eta$ denote the distribution over noisy examples.

For any function $f : \{0,1\}^n \to [-1,1]$,

$$\begin{aligned}\mathbf{E}_{(x,y)\sim\mathcal{N}^\eta}[|f(x)-y|] &= (1-\eta)\mathbf{E}_{x\sim\mathcal{U}}[|f(x)-\chi_S(x)|] + \eta\mathbf{E}_{x\sim\mathcal{U}}[|f(x)+\chi_S(x)|] \\ &= (1-\eta)\mathbf{E}_{x\sim\mathcal{U}}[\chi_S(x)(\chi_S(x)-f(x))] + \eta\mathbf{E}_{x\sim\mathcal{U}}[\chi_S(x)(\chi_S(x)+f(x)) \\ &= 1 + (1-2\eta)\hat{f}(S). \end{aligned} \tag{5}$$

This implies that

$$\mathbf{E}_{(x,y)\sim\mathcal{N}^\eta}[|f_S(x)-y|] = 1 + (1-2\eta)\widehat{f_S}(S) \geq 1 + (1-2\eta)\gamma.$$

By the agnostic property of $\mathcal{A}$ with $\epsilon = (1-2\eta)\gamma/2$, the returned hypothesis h must satisfy

$$\mathbf{E}_{(x,y)\sim\mathcal{N}^\eta}[|h(x)-y|] \geq 1 + (1-2\eta)\gamma - (1-2\eta)\gamma/2 \geq 1 + (1-2\eta)\gamma/2.$$

By equation (5) this implies that $\hat{h}(S) \geq \gamma/2$.

We can now use the algorithm of Goldreich and Levin (1989) (or a similar one) algorithm to find all sets with a Fourier coefficient of at least $\gamma/4$ (with accuracy of $\gamma/8$). This can be done in time polynomial in n and $1/\gamma$ and will give a set of coefficients of size at most $O(1/\gamma^2)$ which contains S. By testing each coefficient in this set on $O((1-2\eta)^{-2}\log(1/\gamma))$ random examples and choosing the one with the best agreement we find S. □

We will now show that there exist monotone symmetric submodular functions that have high correlation with the parity functions (the proof is in Appendix C).

Lemma 6.3 (Correlation of Monotone Submodular Functions with Parities)**.** *Let $S \subseteq [n]$ such that $|S| = s$ for some $s \in [n]$. Then, there exists a* monotone *symmetric submodular function $H_S : \{0,1\}^n \to [0,1]$ such that H_S depends only on coordinates in S and $|\langle \chi_S, H_S \rangle| = \Omega(s^{-3/2})$.*

Combining this result with Lemma 6.2, we now obtain the following reduction of SLPN to agnostically learning monotone submodular functions:

Theorem 6.4 (Theorem 1.8 restated)**.** *If there exists an algorithm that agnostically learns all monotone submodular functions with range $[0,1]$ to ℓ_1 error of $\epsilon > 0$ in time $T(n, 1/\epsilon)$ then there exists an algorithm that learns $(\epsilon^{-2/3})$-sparse parities with noise of rate $\eta < 1/2$ in time $\mathrm{poly}(n, 1/(\epsilon(1-2\eta))) + 2T(n, c/\epsilon(1-2\eta))$ for some fixed constant c.*

Proof. Consider all the monotone submodular functions R_S for every $S \subseteq [n]$, $|S| \leq k = \epsilon^{-2/3}$. Then, $|\langle \chi_S, H_S \rangle| = \Omega(k^{-3/2}) = \Omega(\epsilon)$ by Lemma 6.3. Thus, using $\gamma = \Omega(\epsilon)$ in Lemma 6.2 we obtain the claim. □

6.2 Information-Theoretic Lower Bound for PAC-learning Submodular Functions

In this section we show that any algorithm that PAC-learns monotone submodular functions to accuracy ϵ must use $2^{\Omega(\epsilon^{-2/3})}$ examples. The idea is to show that the problem of learning the class all boolean functions on k variables to any constant accuracy can be reduced to the problem of learning submodular functions on $2t = k + \lceil \log k \rceil + O(1)$ variables to accuracy $O(\frac{1}{t^{3/2}})$. Any algorithm that learns the class of all boolean functions on k variables to accuracy $1/4$ requires at least $\Omega(2^k)$ bits of information. In particular at least that many random examples or value queries are necessary.

Before we go on the present the reduction, we need to make a quick note regarding a slight abuse of notation: In the lemma below, we will encounter uniform distributions on hypercubes of two different dimensions. We will, however, still represent uniform distributions on either of them by $\mathcal{U}$ (with the meaning clear from the context).

Lemma 6.5. *Let $f : \{0,1\}^k \to \{0,1\}$ be any boolean function. Let $t > 0$ be such that $\binom{2t}{t} \geq 2^k > \binom{2t-2}{t-1}$ (thus $4 \cdot 2^k > \binom{2t}{t} \geq 2^k$). There exists a monotone submodular function $h : \{0,1\}^{2t} \to [0,1]$ such that:*

1. *h can be computed at any point $x \in \{0,1\}^{2t}$ in at most a single query to f and in time $O(t)$.*
2. *Let $\alpha = \frac{2^k \cdot \sqrt{t}}{2^{2t}} = \theta(1)$. Given any function $g : \{0,1\}^{2t} \to \mathbb{R}$ that approximates h, that is, $\mathbf{E}_{x \sim \mathcal{U}}[|h(x) - g(x)|] \leq \alpha \cdot \frac{\epsilon}{8t^{3/2}}$, there exists a boolean function $\tilde{f} : \{0,1\}^k \to \{0,1\}$ such that $\mathbf{E}_{x \sim \mathcal{U}}[|\tilde{f}(x) - f(x)|] \leq \epsilon$ and $\tilde{f}$ can be computed at any point $x \in \{0,1\}^k$, with a single query to g and in time $O(t)$.*

Proof. We first give a construction for the function h. It will be convenient first to define another function $\tilde{h} : \{0,1\}^{2t} \to [0,1]$ and then modify it to obtain h. Recall that for any x and $S \subseteq [2t]$, $w_S(x) = \sum_{i \in S} \frac{1}{2}(x_i + 1)$. The function $\tilde{h}$ would be the same as the function H_S defined in the proof of Lemma 6.3.

$$\tilde{h}(x) = \begin{cases} w_{[2t]}(x)/t & w_{[2t]}(x) \leq t \\ 1 & w_{[2t]}(x) > t \end{cases}$$

We will now define h using $\tilde{h}$ and f. The key idea is that even if we lower the value of $\tilde{h}$ at any x with $w_{[2t]}(x) = k$ by $\frac{1}{2t}$, the resulting function remains submodular. Thus, we embed the boolean function h by modifying the values of $\tilde{h}$ at only the points in the middle layer ($w_{[2t]}(x) = t$).

Let $s = \binom{2t}{t}$. Let $M_{2t} = \{x \in \{0,1\}^{2t} \mid w_{[2t]}(x) = t\}$ and $M_k = \{y \in \{0,1\}^k\}$ and $s \geq 2^k$. Let $\beta : M_k \to M_{2t}$ be an injective map of M_k into M_{2t} such that both β and β^{-1} (whenever it exists) can be computed in time $O(t)$ at any given point. Such a map exists, as can be seen by imposing lexicographic ordering on M_{2t} and M_k and defining $\beta(x)$ for $x \in M_{2t}$ to be the element in M_k with the same position in the ordering as that of x. For each $x \in \{0,1\}^{2t}$, let h be defined by:

$$h(x) = \begin{cases} \tilde{h}(x) & w_{[2t]}(x) \neq t \\ (1 - \frac{1}{2t}) & w_{[2t]}(x) = t, \beta^{-1}(x) \text{ exists and } f(\beta^{-1}(x)) = 0 \\ 1 & w_{[2t]}(x) = t, \beta^{-1}(x) \text{ exists and } f(\beta^{-1}(x)) = 1 \\ 1 & \text{otherwise} \end{cases}$$

Notice that given any $x \in \{0,1\}^{2t}$ the value of $h(x)$ can be computed by a single query to f. Further, observe that $\tilde{h}$ is monotone and h is obtained by modifying $\tilde{h}$ only on points in M_{2t} and by at most $\frac{1}{2t}$, which ensures that for any $x \leq y$ such that $w_{[2t]}(x) < w_{[2t]}(y)$, $h(x) \leq h(y)$. Moreover, M_{2t} forms an antichain in the partial order on $\{0,1\}^n$ and thus no two points in M_{2t} are comparable. This proves that h is monotone. Suppose, now that $g : \{0,1\}^{2t} \to \mathbb{R}$ is such that $\mathbf{E}_{x\sim\mathcal{U}}[|h(x) - g(x)|] \leq \alpha \cdot \frac{\epsilon}{8t^{3/2}}$.

Define $g_b : \{0,1\}^{2t} \to \{0,1\}$ so that

$$\forall x \in \{0,1\}^{2t},\, g_b(x) = sign\left(g(x) - (1 - (1/4t))\right).$$

Finally, let $\tilde{f} : \{0,1\}^k \to \{0,1\}$ be such that for every $x \in \{0,1\}^k$ $\tilde{f}(x) = g_b(\beta(x))$.

Now, $\mathbf{E}_{x\sim\mathcal{U}}[|\tilde{f}(x) - f(x)|] = 2\Pr_{x\sim\mathcal{U}}[\tilde{f}(x) \neq f(x)]$. For any $x \in \{0,1\}^k$,

$$\tilde{f}(x) \neq f(x) \Leftrightarrow |g(\beta(x)) - h(\beta(x))| \geq \frac{1}{4t}.$$

Using that $\Pr_{y\sim\mathcal{U}}[\beta^{-1}(y) \text{ exists }] = \frac{\alpha}{\sqrt{t}}$, we have:

$$\begin{aligned} \mathbf{E}_{y\sim\mathcal{U}}[|g(y) - h(y)|] &\geq \frac{1}{4t} \Pr_{y\sim\mathcal{U}}[\beta^{-1}(y) \text{ exists and } \tilde{f}(\beta^{-1}(y) \neq f(\beta^{-1}(y)] \\ &= \frac{1}{8t}\frac{\alpha}{\sqrt{t}}\mathbf{E}_{x\sim\mathcal{U}}[|\tilde{f}(x) - f(x)|]. \end{aligned}$$

Using $\mathbf{E}_{y\sim\mathcal{U}}[|g(y) - h(y)|] \leq \alpha \cdot \frac{\epsilon}{8\cdot(t)^{3/2}}$, we have: $\mathbf{E}_{x\sim\mathcal{U}}[|\tilde{f}(x) - f(x)|] \leq \epsilon$.

Finally, we show that h is submodular for any boolean function f. It will be convenient to switch notation and look at input x as the indicator function of the set $S_x = \{x_i \mid x_i = 1\}$. We will verify that for each $S \subseteq [n]$ and $i, j \notin S$,

$$h(S \cup \{i\}) - h(S) \geq h(S \cup \{i,j\}) - h(S \cup \{j\}). \tag{6}$$

Notice that $\tilde{h}$ is submodular, and $h = \tilde{h}$ on every x such that $w_{[2t]}(x) \neq t$. Thus, we only need to check Equation (6) for S, i, j such that $|S| \in \{t-2, t-1, t\}$. We analyze these 3 cases separately:

1. $\boldsymbol{|S| = t-1}$ **:** Notice that $h(S) = \tilde{h}(S) = 1 - (1/t)$ and $h(S \cup \{i,j\}) = \tilde{h}(S \cup \{i,j\}) = 1$. Also observe that for any f, $h(S\cup\{i\})$ and $h(S\cup\{j\})$ are at least $(1-\frac{1}{2t})$. Thus, $h(S\cup\{i\})+h(S\cup\{j\}) \geq 2 - \frac{1}{t} = h(S) + h(S \cup \{i,j\})$.

2. $\boldsymbol{|S| = t-2}$ **:** In this case, $h(S) = (1 - (2/t))$ and $h(S \cup \{i\}) = h(S \cup \{j\}) = (1 - (1/t))$. In this case, the maximum value for any f, of $h(S \cup \{i,j\}) = 1$. Thus,

$$h(S) + h(S \cup \{i,j\}) \leq 2 - (2/t) = h(S \cup \{i\}) + h(S \cup \{j\}).$$

3. $|\boldsymbol{S}| = \boldsymbol{t}$ **:** Here, $h(S \cup \{i\}) = h(S \cup \{j\}) = h(S \cup \{i,j\}) = 1$. The maximum value of $h(S)$ for any f is 1. Thus,
$$h(S) + h(S \cup \{i,j\}) \leq 2 = h(S \cup \{i\}) + h(S \cup \{j\}).$$

This completes the proof that h is submodular. □

We now have the following lower bound on the running time of any learning algorithm (even with value queries) that learns monotone submodular functions.

Theorem 6.6 (Theorem 1.7 restated)**.** *Any algorithm that PAC learns all monotone submodular functions with range* $[0,1]$ *to* ℓ_1 *error of* $\epsilon > 0$ *requires* $2^{\Omega(\epsilon^{-2/3})}$ *value queries to* f.

Proof. We borrow notation from the statement of Lemma 6.5 here. Given an algorithm that PAC learns monotone submodular functions on $2t$ variables, we describe how one can obtain a learning algorithm for all boolean function on k variables with accuracy $1/4$. Given an access to a boolean function $f : \{0,1\}^k \to \{0,1\}$, we can translate it into an access to a submodular function h on $2t$ variables with an overhead of at most $O(t) = O(k)$ time using Lemma 6.5. Using the PAC learning algorithm, we can obtain a function $g : \{0,1\}^{2t} \to \mathbb{R}$ that approximates h within an error of at most $\alpha \cdot \frac{1}{8t^{3/2}}$ and Lemma 6.5 shows how to obtain $\tilde{f}$ from g with an overhead of at most $O(t) = O(k)$ time such that $\tilde{f}$ approximates f within $\frac{1}{4}$. Choose $k = \lceil \epsilon^{-2/3} \rceil$ and t as described in the statement of Lemma 6.5. Now, using any algorithm that learns monotone submodular functions to an accuracy of $\epsilon > 0$ we obtain an algorithm that learns all boolean functions on $k = \lceil \epsilon^{-2/3} \rceil$ variables to accuracy $1/4$. □

References

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

Ashwinkumar Badanidiyuru, Shahar Dobzinski, Hu Fu, Robert Kleinberg, Noam Nisan, and Tim Roughgarden. Sketching valuation functions. In *SODA*, pages 1025–1035, 2012.

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

M.F. Balcan, Florin Constantin, Satoru Iwata, and Lei Wang. Learning valuation functions. *Journal of Machine Learning Research - COLT Proceedings*, 23:4.1–4.24, 2012.

Eric Blais, Krzysztof Onak, Rocco Servedio, and Grigory Yaroslavtsev. Concise representations of discrete submodular functions, 2013. Personal communication.

A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. A sharp concentration inequality with applications. *Random Struct. Algorithms*, 16(3):277–292, 2000.

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

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

Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *STOC*, pages 610–618, 2005.

Shaddin Dughmi, Tim Roughgarden, and Qiqi Yan. From convex optimization to randomized mechanisms: toward optimal combinatorial auctions. In *STOC*, pages 149–158, 2011.

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

A. Ehrenfeucht and D. Haussler. Learning decision trees from random examples. *Information and Computation*, 82(3):231–246, 1989.

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

V. Feldman. Attribute efficient and non-adaptive learning of parities and DNF expressions. *Journal of Machine Learning Research*, (8):1431–1460, 2007.

V. Feldman. Distribution-specific agnostic boosting. In *Proceedings of Innovations in Computer Science*, pages 241–250, 2010.

V. Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer System Sciences*, 78(5):1444–1459, 2012.

V. Feldman and P. Kothari. Learning coverage functions. Manuscript, 2013.

V. Feldman, P. Gopalan, S. Khot, and A. Ponuswami. On agnostic learning of parities, monomials and halfspaces. *SIAM Journal on Computing*, 39(2):606–645, 2009.

L. Fleischer, S. Fujishige, and S. Iwata. A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777, 2001.

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

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

Michel X. Goemans, Nicholas J. A. Harvey, Satoru Iwata, and Vahab S. Mirrokni. Approximating submodular functions everywhere. In *SODA*, pages 535–544, 2009.

O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of STOC*, pages 25–32, 1989.

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

Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in gaussian processes. In *ICML*, pages 265–272, 2005.

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

D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992. ISSN 0890-5401.

A. Kalai and V. Kanade. Potential-based agnostic boosting. In *Proceedings of NIPS*, pages 880–888, 2009.

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

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

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

Andreas Krause and Carlos Guestrin. Submodularity and its applications in optimized information gathering. *ACM TIST*, 2(4):32, 2011.

Andreas Krause, Carlos Guestrin, Anupam Gupta, and Jon M. Kleinberg. Near-optimal sensor placements: maximizing information while minimizing communication cost. In *IPSN*, pages 2–10, 2006.

Andreas Krause, Ajit Paul Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9: 235–284, 2008.

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

N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

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

Ryan O'Donnell and Rocco A. Servedio. Learning monotone decision trees in polynomial time. *SIAM J. Comput.*, 37(3):827–844, 2007.

Christos H. Papadimitriou, Michael Schapira, and Yaron Singer. On the hardness of being truthful. In *FOCS*, pages 250–259, 2008.

Maurice Queyranne. A combinatorial algorithm for minimizing symmetric submodular functions. In *Proc. of 6th ACM-SIAM SODA*, pages 98–101, 1995.

Sofya Raskhodnikova and Grigory Yaroslavtsev. Learning pseudo-boolean k-dnf and submodular functions. In *Proceedings of SODA*, 2013.

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

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

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

Jan Vondrák. A note on concentration of submodular functions, 2010. arXiv:1005.2791v1.

A Attribute-efficient Agnostic Learning

In this section we give attribute-efficient versions of two agnostic learning algorithms: (1) the ℓ_2-error agnostic learning of functions with low spectral ℓ_1-norm and (2) ℓ_1-error agnostic learning of (real-valued) decision trees. The algorithms are obtained using a simple combination of existing techniques with attribute-efficient weak agnostic parity learning from (Feldman, 2007). For the first algorithm we are not aware of published details of the analysis even without the attribute-efficiency.

We first state the attribute-efficient weak agnostic parity learning from (Feldman, 2007).

Theorem A.1. *There exists an algorithm* $\mathtt{WP}$*, that given an integer* d*,* $\theta > 0$ *and* $\delta \in (0,1]$*, access to value queries of any* $f : \{0,1\}^n \to [-1,1]$ *such that* $|\hat{f}(S)| \geq \theta$ *for some* S*,* $|S| \leq d$*, with probability at least* $1-\delta$*, returns* S'*, such that* $|\hat{f}(S')| \geq \theta/2$ *and* $|S'| \leq d$*.* $\mathtt{WP}(d,\theta,\delta)$ *runs in* $\tilde{O}\left(nd^2\theta^{-2}\log(1/\delta)\right)$ *time and asks* $\tilde{O}\left(d^2\log^2 n\cdot\theta^{-2}\log(1/\delta)\right)$ *value queries.*

Using $\mathtt{WP}$ we can find a set $\mathcal{S}$ of subsets of $[n]$ such that (1) if $S \in \mathcal{S}$ then $|\hat{f}(S)| \geq \theta/2$ and $|S| \leq d$; (2) if $|\hat{f}(S)| \geq \theta$ and $|S| \leq d$ then $S \in \mathcal{S}$. The first property, implies that $|\mathcal{S}| \leq 4/\theta^2$. With probability $1-\delta$, $\mathcal{S}$ can be found in time polynomial in $1/\theta^2$ and the running time of $\mathtt{WP}(d,\theta,4\delta/\theta^2)$. With probability at least $1-\delta$, each coefficient in $\mathcal{S}$ can be estimated to within $\theta/4$ using a random sample of size $\tilde{O}(\log(1/\delta)/\theta^2)$. This gives the following low-degree version of the Kushilevitz-Mansour algorithm (Kushilevitz and Mansour, 1993).

Theorem A.2. *There exists an algorithm* $\mathtt{AEFT}$*, that given an integer* d*,* $\theta > 0$ *and* $\delta \in (0,1]$*, access to value queries of any* $f : \{0,1\}^n \to [-1,1]$*, with probability at least* $1-\delta$*, returns a function* h *represented by the set of its non-zero Fourier coefficients such that*

1. $\mathtt{degree}(h) \leq d$*;*
2. *for all* $S \subseteq [n]$ *such that* $|\hat{f}(S)| \geq \theta$ *and* $|S| \leq d$*,* $\hat{h}(S) \neq 0$*;*
3. *for all* $S \subseteq [n]$*, if* $|\hat{f}(S)| \leq \theta/2$ *then* $\hat{h}(S) = 0$*;*
4. *if* $\hat{h}(S) \neq 0$ *then* $|\hat{f}(S) - \hat{h}(S)| \leq \theta/4$*.*

$\mathtt{AEFT}(d,\theta,\delta)$ *runs in* $\tilde{O}\left(nd^2\theta^{-2}\log(1/\delta)\right)$ *time and asks* $\tilde{O}\left(d^2\log^2 n\cdot\theta^{-2}\log(1/\delta)\right)$ *value queries.*

We now show that for $\theta = \epsilon^2/(2L)$, $\mathtt{AEFT}$ agnostically learns the class

$$\mathcal{C}_L^d = \{p(x) \mid \|\hat{p}\|_1 \leq L \text{ and } \mathtt{degree}(p) \leq d\}\ .$$

Lemma A.3. *For* $L > 0, \epsilon \in (0,1)$ *and integer* d*, let* $f : \{0,1\}^n \to [-1,1]$ *and* $h :\to \mathbb{R}$ *be functions such that for* $\theta = \epsilon^2/(2L)$*,*

1. $\mathtt{degree}(h) \leq d$*;*

2. *for all* $S \subseteq [n]$ *such that* $|\hat{f}(S)| \geq \theta$ *and* $|S| \leq d$, $\hat{h}(S) \neq 0$;

3. *for all* $S \subseteq [n]$, *if* $|\hat{f}(S)| \leq \theta/2$ *then* $\hat{h}(S) = 0$;

4. *if* $\hat{h}(S) \neq 0$ *then* $|\hat{f}(S) - \hat{h}(S)| \leq \theta/4$.

Then for any $g \in \mathcal{C}_L^d$, $\|f - h\|_2 \leq \|f - g\|_2 + \epsilon$.

Proof. We show that for every $S \subseteq [n]$,

$$(\hat{f}(S) - \hat{h}(S))^2 \leq (\hat{f}(S) - \hat{g}(S))^2 + 2\theta \cdot |\hat{g}(S)| = (\hat{f}(S) - \hat{g}(S))^2 + \frac{\epsilon^2 \cdot |\hat{g}(S)|}{L}. \tag{7}$$

First note that this would immediately imply that

$$\begin{aligned}\|f - h\|_2^2 &= \sum_{S \subseteq [n]} (\hat{f}(S) - \hat{h}(S))^2 \leq \sum_{S \subseteq [n]} (\hat{f}(S) - \hat{g}(S))^2 + \frac{\epsilon^2 \cdot |\hat{g}(S)|}{L} = \|f - g\|_2^2 + \frac{\epsilon^2 \cdot \|\hat{g}\|_1}{L} \\ &\leq \|f - g\|_2^2 + \epsilon^2 \leq (\|f - g\|_2 + \epsilon)^2.\end{aligned}$$

To prove equation (7) we consider two cases. If $\hat{h}(S) = 0$, then either $|S| > d$ or $|\hat{f}(S)| \leq \theta$. In the former case $\hat{g}(S) = 0$ and therefore equation (7) holds. In the latter case:

$$(\hat{f}(S) - \hat{h}(S))^2 = (\hat{f}(S))^2 \leq (\hat{f}(S) - \hat{g}(S))^2 + 2|\hat{f}(S)| \cdot |\hat{g}(S)| \leq (\hat{f}(S) - \hat{g}(S))^2 + 2\theta \cdot |\hat{g}(S)| \,.$$

In the second case (when $\hat{h}(S) \neq 0$), we get that $|\hat{f}(S)| \geq \theta/2$ and $|\hat{f}(S) - \hat{h}(S)| \leq \theta/4$. Therefore, either $|\hat{g}(S)| \leq |\hat{f}(S)|/2$ and then $(\hat{f}(S) - \hat{g}(S))^2 \geq (\hat{f}(S))^2/4 \geq \theta^2/16$ or $|\hat{g}(S)| \geq |\hat{f}(S)|/2 \geq \theta/4$ and then $2\theta \cdot |\hat{g}(S)| \geq \theta^2/2$. In both cases,

$$(\hat{f}(S) - \hat{h}(S))^2 \leq \frac{\theta^2}{16} \leq (\hat{f}(S) - \hat{g}(S))^2 + 2\theta \cdot |\hat{g}(S)| \,.$$

□

Theorem 5.3 is a direct corollary of Theorem A.2 and Lemma A.3.

The proof of Theorem 5.5 relies on agnostic learning of decision trees. We first give an attribute-efficient algorithm for this problem.

Theorem A.4. *Let* $\mathtt{DT}(r)$ *denote the class of all Boolean decision trees of rank-*r *on* $\{0,1\}^n$. *There exists an algorithm* $\mathcal{A}$ *that given* $\epsilon > 0$ *and access to value queries of any* $f : \{0,1\}^n \to \{0,1\}$, *with probability at least* $2/3$, *outputs a function* $h : \{0,1\}^n \to \{0,1\}$, *such that* $\mathrm{Pr}_{\mathcal{U}}[f \neq h] \leq \Delta + \epsilon$, *where* $\Delta = \min_{g \in \mathtt{DT}(r)}\{\mathrm{Pr}_{\mathcal{U}}[f \neq g]\}$. *Further,* $\mathcal{A}$ *runs in time* $\mathrm{poly}(n, 2^r, 1/\epsilon)$ *and uses* $\mathrm{poly}(\log n, 2^r, 1/\epsilon)$ *value queries.*

Proof. We first use Theorem 4.1 to reduce the problem of agnostic learning of decision trees of rank at most r to the problem of agnostic learning of decision trees of depth $\frac{5}{2}(r + \log(2/\epsilon))$ with error parameter $\epsilon/2$. In (Feldman, 2010) and (Kalai and Kanade, 2009) it is shown that a distribution-specific agnostic boosting algorithm reduces the problem of agnostic learning decision trees of size s with error $\epsilon' = \epsilon/2$ to that of weak agnostic learning of decision trees invoked $O(s^2/\epsilon'^2)$ times. It was also shown in those works that agnostic learning of parities with error of $\epsilon'/(2s)$ gives the necessary weak agnostic learning of decision trees. Further, as can be easily seen from the proof, for decision trees of depth $\leq d$ it is sufficient to agnostically

learn parities of degree $\leq d$. In our case the size of the decision tree is $\leq 2^d = (2^{r+1}/\epsilon)^{5/2}$. We can use WP algorithm with error parameter $\epsilon'/(2s) \geq \epsilon^{7/2}/2^{\frac{5r}{2}+5}$ and degree d, to obtain weak agnostic learning of decision trees in time $\mathrm{poly}(n, 2^r, 1/\epsilon)$ and using $\mathrm{poly}(\log n, 2^r, 1/\epsilon)$ value queries. This implies that agnostic learning of decision trees can be achieved in time $\mathrm{poly}(n, 2^r, 1/\epsilon)$ and using $\mathrm{poly}(\log n, 2^r, 1/\epsilon)$ value queries. □

From here we can easily obtain an algorithm for agnostic learning of rank-r decision trees with real-valued constants from $[0, 1]$. We obtain it by using a simple argument (see (Feldman and Kothari, 2013) for a simple proof) that reduces learning of a real-valued function g to learning of boolean functions of the form $g_\theta(x) =$ "$g(x) \geq \theta$" (note that every $g : \{0,1\}^n \to [0,1]$, is ϵ-close (in ℓ_1 distance) to $g'(x) = \sum_{i \in \lfloor 1/\epsilon \rfloor} g_{i\epsilon}(x)$). We now observe that if g can be represented as a decision tree of rank r, then for every θ, g_θ can be represented as a decision tree of rank r. Therefore this reduction implies that agnostic learning of Boolean rank-r decision trees gives agnostic learning of $[0, 1]$-valued rank-r decision trees. The reduction runs the Boolean version $2/\epsilon$ times with accuracy $\epsilon/2$ and yields the proof of Theorem 5.4.

B Learning Pseudo-Boolean Submodular Functions

In a recent work, Raskhodnikova and Yaroslavtsev (2013) consider learning and testing of submodular functions taking values in the range $\{0, 1, \ldots, k\}$. The error of a hypothesis in their framework is the probability that the hypothesis disagrees with the unknown function (hence it is referred to as *pseudo-Boolean*). For this restriction they give a $\mathrm{poly}(n) \cdot k^{O(k \log k/\epsilon)}$-time PAC learning algorithm using value queries.

As they observed, error ϵ in their model can also be obtained by learning the function scaled to the range $\{0, 1/k, \ldots, 1\}$ with ℓ_1 error of ϵ/k (since for two functions with that range $\mathbf{E}[|f - h|] \leq \epsilon/k$ implies that $\Pr[f \neq h] \leq \epsilon$). Therefore our structural results can also be interpreted in their framework directly. We now show that even stronger results are implied by our technique.

The first observation is that a $\frac{1}{k+1/3}$-Lipschitz function with the range $\{0, 1/k, \ldots, 1\}$ is a constant. Therefore Theorem 3.1 implies an exact representation of submodular functions with range $\{0, 1, \ldots, k\}$ by decision trees of rank $\leq \lfloor 2k + 2/3 \rfloor = 2k$ with constants from $\{0, 1/k, \ldots, 1\}$ in the leafs. We note that this representation is incomparable to $2k$-DNF representation which is the basis of results in (Raskhodnikova and Yaroslavtsev, 2013).

We can also directly combine Theorems 3.1 and 4.1 to obtain the following analogue of Corollary 1.3.

Theorem B.1. *Let $f : \{0,1\}^n \to \{0, 1, \ldots, k\}$ be a submodular function and $\epsilon > 0$. There exists a $\{0, 1, \ldots, k\}$-valued decision tree T of depth $d = 5(k + \log(1/\epsilon))$ such that $\Pr_{\mathcal{U}}[T \neq f] \leq \epsilon$. In particular, T depends on at most $2^{5k}/\epsilon^5$ variables and $\|\hat{T}\|_1 \leq 2k \cdot 2^{5k}/\epsilon^5$.*

These results improve on the spectral norm bound of $k^{O(k \log k/\epsilon)}$ from (Raskhodnikova and Yaroslavtsev, 2013). In a follow-up (independent of this paper) work Blais et al. (2013) also obtained an approximation of discrete submodular functions by juntas. They prove that every submodular function f of range of size k is ϵ-close to a function of $(k \log(k/\epsilon))^{O(k)}$ variables and give an algorithm for testing submodularity using $(k \log(1/\epsilon))^{\tilde{O}(k)}$ value queries. Note that our bound has a better dependence on k but worse on ϵ (the bounds have the same order when $\epsilon = k^{-k}$).

As in the general case, these structural results can be used to obtain learning algorithms in this setting. It is natural to require that learning algorithms in this setting output a $\{0, 1, \ldots, k\}$-valued hypothesis. We observe that the algorithm in Theorem 5.4 can be easily modified to return a $\{0, 1/k, \ldots, 1\}$-valued function

when it is applied for learning $\{0, 1/k, \ldots, 1\}$-valued functions. This is true since the proof of Theorem 5.4 (see Section A discretizes the target function and reduces the problem to learning of Boolean functions. $\{0, 1/k, \ldots, 1\}$-valued functions are already discretized. With this exact discretization the output of the agnostic algorithm is a sum of k Boolean hypotheses, and in particular is a $\{0, 1/k, \ldots, 1\}$-valued function. This immediately leads to the following algorithm for agnostic learning of $\{0, 1, \ldots, k\}$-valued submodular functions.

Theorem B.2. *Let $\mathcal{C}_s^k$ denote the class of all submodular functions from $\{0,1\}^n$ to $\{0, 1, \ldots, k\}$. There exists an algorithm $\mathcal{A}$ that given $\epsilon > 0$ and access to value queries of any $f : \{0,1\}^n \to \{0, 1, \ldots, k\}$, with probability at least $2/3$, outputs a function h with the range in $\{0, 1, \ldots, k\}$, such that $\mathbf{E}_{\mathcal{U}}[|f - h|] \leq \Delta + \epsilon$, where $\Delta = \min_{g \in \mathcal{C}_s^k}\{\mathbf{E}_{\mathcal{U}}[|f - g|]\}$. Further, $\mathcal{A}$ runs in time $\mathrm{poly}(n, 2^k, 1/\epsilon)$ and uses $\mathrm{poly}(\log n, 2^k, 1/\epsilon)$ value queries.*

This improves on $\mathrm{poly}(n) \cdot k^{O(k \log k/\epsilon)}$-time and queries algorithm with the same guarantees which is implied by the spectral bounds in (Raskhodnikova and Yaroslavtsev, 2013). We remark that the guarantee of this algorithm implies PAC learning with disagreement error (since for integer valued hypotheses ℓ_1-error upper-bounds the disagreement error). At the same time the guarantee is not agnostic in terms of the disagreement error[2] (but only for ℓ_1-error).

The structural results also imply that when adapted to this setting our PAC learning algorithm in Theorem 1.5 leads to the following PAC learning algorithm in this setting.

Theorem B.3. *There exists an algorithm $\mathcal{A}$ that given $\epsilon > 0$ and access to random uniform examples of any $f \in \mathcal{C}_s^k$, with probability at least $2/3$, outputs a function h, such that $\Pr_{\mathcal{U}}[f \neq h] \leq \epsilon$. Further, $\mathcal{A}$ runs in time $\tilde{O}(n^2) \cdot 2^{O(k^2 + \log^2(1/\epsilon))}$ and uses $2^{O(k^2 + \log^2(1/\epsilon))} \log n$ examples.*

For learning from random examples alone, previous structural results imply only substantially weaker bounds: ($\mathrm{poly}(n^k, 1/\epsilon)$ in (Raskhodnikova and Yaroslavtsev, 2013)).

Finally, we show that the combination of approximation by a junta and exact representation by a decision tree lead to a proper PAC learning algorithm for pseudo-Boolean submodular functions in time $\mathrm{poly}(n) \cdot 2^{O(k^2 + k \log(1/\epsilon))}$ using value queries. Note that, for the general submodular functions our results imply only a doubly-exponential time algorithm (with singly exponential number of random examples).

Theorem B.4. *Let $\mathcal{C}_s^k$ denote the class of all submodular functions from $\{0,1\}^n$ to $\{0, 1, \ldots, k\}$. There exists an algorithm $\mathcal{A}$ that given $\epsilon > 0$ and access to value queries of any $f \in \mathcal{C}_s^k$, with probability at least $2/3$, outputs a* submodular *function h, such that $\Pr[f \neq h] \leq \epsilon$. Further, $\mathcal{A}$ runs in time $\mathrm{poly}(n, 2^{k^2 + k \log 1/\epsilon})$ and uses $\mathrm{poly}(\log n, 2^{k^2 + k \log 1/\epsilon})$ value queries.*

Proof Outline: In the first step we identify a small set of variables J such that there exists a function that depends only on variables indexed by J and is $\epsilon/3$ close to f. This can be achieved (with probability at least $2/3$) by using the algorithm in Lemma 1.4 (with bounds adapted to this setting) to obtain a set of size $\mathrm{poly}(2^k/\epsilon)$. Now let $\mathcal{U}_J$ represent a uniform distribution over $\{0,1\}^J$ and $\mathcal{U}_{\bar{J}}$ represent the uniform distribution over $\bar{J} = [n] \setminus J$. Let g be the function that depends only on variables in J and is $\epsilon/3$ close to f. Then,

$$\Pr_{\mathcal{U}}[f(x) \neq g(x)] = \mathbf{E}_{z \sim \mathcal{U}_{\bar{J}}}\left[\Pr_{y \sim \mathcal{U}_J}[f(y,z) \neq g(y,\bar{0})]\right] \leq \epsilon/3\,.$$

[2] In (Raskhodnikova and Yaroslavtsev, 2013) it was mistakenly claimed that the application of the algorithm of Gopalan et al. (2008) gives agnostic guarantee for the disagreement error.

By Markov's inequality, this means that with probability at least $1/2$ over the choice of z from $\{0,1\}^{\bar{J}}$, $\Pr_{y\sim\mathcal{U}_J}[f(y,z) \neq g(y,\bar{0})] \leq 2\epsilon/3$ and hence $\Pr_{y\sim\mathcal{U}_J, w\sim\mathcal{U}_{\bar{J}}}[f(y,z) \neq f(y,w)] \leq \epsilon$. In other words, a random restriction of variables outside of J gives, with probability at least $1/2$, a function that is ϵ-close to f. As before we observe that a restriction of a submodular function is a submodular function itself. We therefore can choose z randomly and then run the decision tree representation construction algorithm on $f(y,z)$ as a function of y described in the proof of Theorem 3.1. It is easy to see that the running time of the algorithm is essentially determined by the size of the tree. A tree of rank $2k$ over $|J|$ variables has size of at most $|J|^{2k}$ (Ehrenfeucht and Haussler, 1989). Therefore with probability at least $2/3 \cdot 1/2 = 1/3$, in time $\mathrm{poly}(n, 2^{k^2+k\log 1/\epsilon})$ and using $\mathrm{poly}(\log n, 2^{k^2+k\log 1/\epsilon})$ value queries we will obtain a submodular function which is ϵ-close to f. As usual the probability of success can be easily boosted to $2/3$ by repeating the algorithm 3 times and testing the hypothesis. □

C Proof of Lemma 6.3

Since the functions we are dealing with are going to be symmetric, we make the convenient definition of weight of any $x \in \{0,1\}^n$. For any $x \in \{0,1\}^n$, the weight of x over a subset $S \subseteq [n]$ of coordinates is defined as $w_S(x) = \sum_{i\in S} x_i$.

Our correlation bounds for monotone symmetric submodular functions will depend on the following well-known observation which we state without proof.

Fact C.1 (Symmetric Submodular Functions from Concave Profiles)**.** *Let $p : \{0,1,\ldots,n\} :\to [0,1]$ be any function such that,*

$$\forall 0 \leq i \leq n-2,\ p(i+1) - p(i) \geq p(i+2) - p(i+1).$$

Let $f_p : \{0,1\}^n \to [0,1]$ be a symmetric function such that $f_p(x) = p(w_{[n]}(x))$. Then f is submodular.

Remark C.2. *Observe that for any submodular function $f : \{0,1\}^S \to [0,1]$, the correlation with the parity χ_S depends only on the profile of f, $p_f : \{0,1,\ldots,n\} \to [0,1]$,*

$$\forall i,\ p_f(i) = \frac{1}{\binom{n}{i}} \sum_{x:w_S(x)=i} f(x).$$

That is, if $\tilde{f} : \{0,1\}^S \to [0,1]$ is defined by $\tilde{f}(x) = p_f(w_S(x))$ for every $x \in \{0,1\}^n$, then $\langle f, \chi_S\rangle = \langle \tilde{f}, \chi_S\rangle$. Thus for finding submodular functions with large correlation with a given parity, it is enough to focus on symmetric submodular functions.

We will need the following well-known formula for the partial sum of binomial coefficients in our correlation bounds.

Fact C.3 (Alternating Binomial Partial Sum)**.** *For every $n, r, k \in \mathbb{N}$,*

$$\sum_{j=0}^{r} (-1)^j \binom{n}{j} = (-1)^r \binom{n-1}{r}$$

Proof of Lemma 6.3. Notice that the parity on any subset $S \subseteq [n]$ of variables at any input $x \in \{0,1\}^n$ is computed by $\chi_S(x) = (-1)^{w_S(x)}$. We will now define a symmetric submodular function $R_S : \{0,1\}^S \to$

$[0,1]$ and then modify it to construct a monotone symmetric submodular function $H_S : \{0,1\}^S \to [0,1]$ that has the required correlation with the associated parity χ_S. It is easy to verify that the natural extension of R_S and H_S to $\{0,1\}^n$(from $\{0,1\}^S$), that just ignores all the coordinates outside S, is submodular and thus it is enough to construct functions on $\{0,1\}^S$.

The definition of R_S will vary based on the cardinality of S. If S is such that $s = 2k$ for some $k \in \mathbb{N}$, let R_S for each $S \subseteq [n]$ be defined as follows:

$$R_S(x) = \begin{cases} \frac{w_S(x)}{k}, & w_S(x) \leq k \\ 1 - \frac{w_S(x)-k}{k}, & w_S(x) > k \end{cases}$$

On the other hand, if S is such that $s = 2k-1$ for some $k \in \mathbb{N}$, define:

$$R_S(x) = \begin{cases} \frac{w_S(x)}{k-1}, & w_S(x) \leq k-1 \\ 1 - \frac{w_S(x)-k+1}{k-1}, & w_S(x) \geq k \end{cases}$$

Notice that with this definition, $R_S : \{0,1\}^n \to [0,1]$ and has its maximum value exactly equal to 1. Further, since R_S can be seen to be defined by a concave profile, Fact C.1 guarantees that R_S is submodular. We will now compute the correlation of χ_S with R_S. We will first deal with the case when $|S|$ is even.

Let $s = 2k$ for some $k \in \mathbb{N}$.

$$\begin{aligned}
\langle R_S, \chi_S \rangle &= \frac{1}{2^{2k}} \sum_{x \in \{0,1\}^{2k}} R_S(x)\chi_S(x) \\
&= \frac{1}{2^{2k}} \cdot \sum_{i=0}^{k} \binom{2k}{i} (-1)^i \frac{i}{k} + \sum_{i=k+1}^{2k} \binom{2k}{i} (-1)^i (1 - \frac{i-k}{k}) \\
&\text{Substituting } j = 2k - i \\
&= \frac{1}{2^{2k}} \cdot \sum_{i=0}^{k} \binom{2k}{i} (-1)^i \frac{i}{k} + \sum_{j=0}^{k-1} \binom{2k}{j} (-1)^j \frac{j}{k} \\
&= 2\left(\frac{1}{2^{2k}} \cdot \frac{1}{k} \sum_{i=0}^{k} \binom{2k}{i} (-1)^i \cdot i \right) - (-1)^k \cdot \frac{1}{2^{2k}} \binom{2k}{k} \\
&= 2\left(\frac{1}{2^{2k}} \cdot \frac{1}{k} \cdot 2k \cdot \sum_{i=1}^{k} \binom{2k-1}{i-1} (-1)^i \right) - (-1)^k \cdot \frac{1}{2^{2k}} \binom{2k}{k} \\
&\text{Using the partial sum formula from Fact C.3 gives:} \\
\langle R_S, \chi_S \rangle &= (-1)^k \cdot \frac{2}{2^{2k}} \cdot \frac{1}{2k-1} \binom{2k-1}{k}
\end{aligned}$$

Now suppose $s = 2k-1$ for some $k \in \mathbb{N}$.

$$\begin{aligned}
\langle R_S, \chi_S \rangle &= \frac{1}{2^{2k-1}} \sum_{x \in \{0,1\}^{2k-1}} R_S(x)\chi_S(x) \\
&= \frac{1}{2^{2k-1}} \cdot \sum_{i=0}^{k-1} \binom{2k-1}{i} (-1)^i \frac{i}{k-1} + \sum_{i=k}^{2k-1} \binom{2k-1}{i} (-1)^i (1 - \frac{i-k+1}{k-1}) \\
&\text{Substituting } j = 2k-1-i \\
&= \frac{1}{2^{2k-1}} \cdot \sum_{i=0}^{k} \binom{2k}{i} (-1)^i \frac{i}{k} - \sum_{j=0}^{k-1} \binom{2k-1}{j} (-1)^j \frac{j-1}{k-1} \\
&= \frac{1}{2^{2k-1}} \frac{1}{k-1} \cdot \sum_{j=0}^{k-1} \binom{2k-1}{j} (-1)^j \\
&\text{Again, using the partial sum formula from Fact C.3 gives:} \\
\langle R_S, \chi_S \rangle &= (-1)^{k+1} \cdot \frac{1}{2^{2k-1}} \cdot \frac{1}{k-1} \binom{2k-2}{k-1}
\end{aligned}$$

In either case, we now obtain that $|\langle R_S, \chi_S \rangle| = \Omega(k^{\frac{-3}{2}}) = \Omega(s^{\frac{-3}{2}})$.

For the remaining part of the proof, we need to define the function H_S. We obtain H_S by a natural "monotonization" of R_S. Thus, if $s = 2k$, let H_S be defined as:

$$H_S(x) = \begin{cases} \frac{w_S(x)}{k}, & w_S(x) \leq k \\ 1 & w_S(x) > k \end{cases}$$

On the other hand, if S is such that $s = 2k-1$ for some $k \in \mathbb{N}$, define:

$$R_S(x) = \begin{cases} \frac{w_S(x)}{k-1}, & w_S(x) \leq k-1 \\ 1 & w_S(x) \geq k \end{cases}$$

Notice again that $H_S : \{0,1\}^S \to [0,1]$ and H_S is submodular by Fact C.1. To obtain a lower bound on $|\langle \chi_S, H_S \rangle|$, H_S can be seen as the average of a monotone linear function and R_S, that is, if $s = 2k$, $\forall x$, $H_S(x) = \frac{1}{2}(R_S(x) + \frac{w_S(x)}{k})$ and if $s = 2k-1$, $\forall x$, $H_S(x) = \frac{1}{2}(R_S(x) + \frac{w_S(x)}{k-1})$. It is now easy to obtain a lower bound on the correlation of χ_S with H_S.

For $s = 2k$,

$$\langle \chi_S, H_S \rangle = \frac{1}{2}\langle \chi_S, R_S \rangle + \frac{1}{2}\langle \chi_S, \frac{w_S}{k} \rangle.$$

For $s = 2k-1$,

$$\langle \chi_S, H_S \rangle = \frac{1}{2}\langle \chi_S, R_S \rangle + \frac{1}{2}\langle \chi_S, \frac{w_S}{k-1} \rangle.$$

Finally, observe that for any $s = |S|$, $\langle \chi_S, w_S(x) \rangle = \sum_{i=0}^{s} \binom{s}{i}(-1)^i \cdot i = s \sum_{i=0}^{s} \binom{s-1}{i-1}(-1)^i \cdot i = 0$. This immediately yields the required correlation.

□