Learning Set Functions with Limited Complementarity

Hanrui Zhang
Computer Science Department
Duke University
Durham, NC 27705
hrzhang@cs.duke.edu

Abstract

We study PMAC-learning of real-valued set functions with limited complementarity. We prove, to our knowledge, the first nontrivial learnability result for set functions exhibiting complementarity, generalizing Balcan and Harvey's result for submodular functions. We prove a nearly matching information theoretical lower bound on the number of samples required, complementing our learnability result. We conduct numerical simulations to show that our algorithm is likely to perform well in practice.

Introduction

A central problem in economics and algorithmic game theory is to price items. Intuitively, a seller would like to set the highest price such that the customer would still buy, which requires decent understanding of the customer's valuation. When there are multiple items which can be sold in any combination, the valuation is usually modeled as a set function with the items being the ground set. That is, the valuation function of the customer maps each subset of the items to her utility when she gets the subset. To be able to better price the items for maximum profit, one need to learn the customer's valuation function. Set functions are also used to model influence propogation in social networks (Kempe, Kleinberg, and Tardos 2003), and for solving clustering problems (Narasimhan and Bilmes 2007). In all these scenarios, learning the corresponding set function plays an essential part in solving the problem.

There is a rich body of research on learning of set functions, e.g. (Balcan and Harvey 2011; Balcan et al. 2012; Lin and Bilmes 2012; Bach and others 2013). All of these results focus on an important class of monotone set functions — *complement-free* set functions. Such set functions model the natural property of diminishing returns, and are generally considered much easier to tackle than general monotone set functions. For example, various optimization problems admit efficient constant factor approximations when the set function involved is complement-free or submodular (which is stronger than complement-free) (Nemhauser and Wolsey 1978; Vondrák 2008; Feige 2009), while for general monotone functions the best possible approximation ratio can be arbitrarily large.

However, in real-world scenarios, it is common that a valuation function exhibits *limited complementarity*. For example, a pen is useful only if accompanied by paper to write on. Paper therefore complements pens. This complementarity to pens is limited, in the sense that owning items other than paper, like computers, is unlikely to make pens more valuable. So in the above example, complementarity exists only between pens and paper. One significant real-world example of limited complementarity is the spectrum auctions, where the rights to use specific bands in specific regions are sold. The complementarity there lies in the fact that a buyer would like the same band in neighboring regions (say states). Since there are only 50 states, one would naturally cosider the degree of complementarity limited. More motivating everyday examples of limited complementarity can be found in (Feige et al. 2015; Eden et al. 2017; Chen, Teng, and Zhang 2019).

In the past decade, there has been a growing interest in studying set functions with limited complementarity, especially in the combinatorial optimization and algorithmic game theory communities. In particular, recent results seem to suggest, that there exists smooth transitions from complement-free to completely arbitrary monotone set functions, parametrized by the *degree of complementarity* of the function. The transitions support graceful degrading of the approximation ratio for various combinatorial optimization tasks (Feige and Izsak 2013; Feldman and Izsak 2014; Feige et al. 2015; Chen, Teng, and Zhang 2019), and the revenue and efficiency (measured by the Price of Anarchy) of well-studied simple protocols for combinatorial auctions (Feige et al. 2015; Feldman et al. 2016; Eden et al. 2017; Chen, Teng, and Zhang 2019).

So one natural question arises:

Is there a way to generalize learnability of complement-free set functions to those with limited complementarity, without incurring too much penalty?

In this paper, based on understanding of the underlying combinatorial and statistical structures, we give, to our knowledge, the first nontrivial learnability result for monotone set functions with limited complementarity:

Theorem 1 (Main Theorem (Informal))**.** *Restricted to product distributions, there is an efficient algorithm that* $O(1/\log(\varepsilon))$*-approximately learns any monotone set func-*

tion with fixed degree of complementarity.

The above theorem generalizes a central result of (Balcan and Harvey 2011) beyond complement-free functions. We also complement our result by a nearly matching information theoretical lower bound. We conduct numerical simulations to show that our algorithm is likely to perform well in practice.

Define the *marginal* of S given T, denoted by $f(S|T)$, to be $f(S|T) := f(S \cup T) - f(T)$. Throughout the paper, when we refer to a set function f, unless otherwise specified, we always assume that:

- f has a *ground set* $[n] = \{1, 2, \ldots, n\}$. That is, f maps all subsets of $[n]$, denoted by $2^{[n]}$, to real numbers.
- f is (weakly) *monotone*. That is, for any $S \subseteq T \subseteq [n]$, $f(S) \leq f(T)$.
- f is 1-*Lipschitz*. That is, for any $S \subseteq [n]$ and $v \in [n]$, $f(v|S) \leq 1$.

PMAC-Learning

To study learnability of real-valued functions, we use the Probably Mostly Approximately Correct (PMAC) model introduced by Balcan and Harvey in (Balcan and Harvey 2011).

Definition 1 (PMAC-Learning (Balcan and Harvey 2011))**.** Let $\mathcal{F}$ be a family of functions with domain $2^{[n]}$. We say that an algorithm $\mathcal{A}$ PMAC-learns $\mathcal{F}$ with approximation factor α, if for any distribution $\mathcal{D}$ over $2^{[n]}$, target function $f^* \in \mathcal{F}$, and for any sufficiently small $\varepsilon \geq 0, \delta \geq 0$, $\mathcal{A}$ takes as input a set of samples $\{(S_i, f^*(S_i)\}_{i \in [\ell]}$ where each S_i is drawn independently from $\mathcal{D}$, and outputs a function $f : 2^{[n]} \to \mathbb{R}$ in $\mathcal{F}$ that satisfies

$$\Pr_{S_1,\ldots,S_\ell \sim \mathcal{D}} \left[\Pr_{S \sim \mathcal{D}} [f(S) \leq f^*(S) \leq \alpha \cdot f(S)] \geq 1 - \varepsilon \right] \geq 1 - \delta,$$

where the number of samples ℓ and the running time of $\mathcal{A}$ are both $\mathrm{poly}(n, 1/\varepsilon, 1/\delta)$.

In words, the definition says the algorithm succeeds with probability $1 - \delta$, upon which it outputs an approximation f of f^* such that with probability $1 - \varepsilon$, $f^*(S)$ is within factor α of $f(S)$. Note that restricted to Boolean-valued functions and letting $\alpha = 1$, PMAC-learning becomes exactly the classic PAC-learning.

Classes of Set Functions

Numerous classes of complement-free set functions have been proposed and studied, among which the following classes are particularly natural and useful: *submodular*, *fractionally subadditive*, and *subadditive* functions. Previous work on learning set functions has been focusing on these classes.

- *Submodular*. A set function f is submodular, if for any $v \in [n]$, $S, T \subseteq [n]$, $f(v|S \cup T) \leq f(v|S)$. The class contains essentially all functions with diminishing marginal returns.
- *Fractionally subadditive (or XOS).* A set function f is fractionally subadditive, if for any $S \subseteq [n]$, $k \in \mathbb{N}$, $T_1, \ldots, T_k \subseteq [n]$, $0 \leq \alpha_1, \ldots, \alpha_k \leq 1$, $f(S) \geq \sum_{i \in [k]} \alpha_i f(T_i)$, as long as the following holds: for any $v \in S$, $\sum_{i \in [k]: v \in T_i} \alpha_i \geq 1$. In other words, if $\{(T_i, \alpha_i)\}_i$ form a fractional cover of S, then the weighted sum of $f(T_i)$'s is no smaller than $f(S)$.
- *Subadditive (or complement-free).* A set function f is subadditive, if for any $S, T \subseteq [n]$, $f(S) + f(T) \geq f(S \cup T)$.

It can be shown that every submodular function is fractionally subadditive, and every fractionally subadditive function is subadditive.

Beyond complement-free functions, several measures of complementarity have been proposed, and the ones particularly helpful for our purposes are the *supermodular degree (SD) hierarchy* and the *supermodular width (SMW) hierarchy*. They build on the concepts of *positive dependency* and *supermodular sets* respectively.

Definition 2 (Positive Dependency (Feige and Izsak 2013))**.** Given a set function f, an element $u \in [n]$ depends positively on $v \in [n]$, denoted by $u \to^+ v$, if there exists $S \subseteq [n] \setminus \{u\}$, such that $f(u|S) > f(u|S \setminus \{v\})$.

Definition 3 (Supermodular Degree Hierarchy (Feige and Izsak 2013))**.** The supermodular degree of a set function f, denoted by $\mathrm{SD}(f)$, is defined to be

$$\mathrm{SD}(f) := \max_u |\{v \mid u \to^+ v\}|.$$

For any $d \in \{0, 1, \ldots, n-1\}$, a function f is in the first d levels of the supermodular degree hierarchy, denoted by $f \in$ SD-d, if $\mathrm{SD}(f) \leq d$.

The definitions essentially say, that u depends positively on v if adding v to some set makes the marginal of u given that set strictly larger, and the supermodular degree of f is then the maximum number of elements on which some particular element positively depends. The degree then naturally categorizes functions into hierarchies.

Definition 4 (Supermodular Set (Chen, Teng, and Zhang 2019))**.** A set $T \subseteq [n]$ is a supermodular set w.r.t. f, if there exists $v \in [n]$ and $S \subseteq [n]$, such that for all $T' \subsetneq T$,

$$f(v|S \cup T) > f(v|S \cup T').$$

Definition 5 (Supermodular Width Hierarchy (Chen, Teng, and Zhang 2019))**.** The supermodular width of a set function f, denoted by $\mathrm{SMW}(f)$, is defined to be

$$\mathrm{SMW}(f) := \max\{|T| \mid T \text{ is a supermodular set}\}.$$

For any $d \in \{0, 1, \ldots, n-1\}$, a function f is in the first d levels of the supermodular width hierarchy, denoted by $f \in$ SMW-d, if $\mathrm{SMW}(f) \leq d$.

That is to say, T is a supermodular set, if given "environment" S, v has a larger marginal given T than given any proper subset of T, and the supermodular width of f is the size of the largest supermodular set.

One can show that the lowest levels of the two hierarchies, SD-0 and SMW-0, coincide. In fact, they are exactly the family of submodular functions. And the highest levels of the two hierarchies, SD-$(n-1)$ and SMW-$(n-1)$, contain all monotone set functions. It can also be shown that:

Proposition 1 ((Chen, Teng, and Zhang 2019)). *For any set function f, $\mathrm{SMW}(f) \leq \mathrm{SD}(f)$. Or equivalently, for any $d \in \{0, 1, \ldots, n-1\}$, SD-$d \subseteq$ SMW-d.*

So the SMW hierarchy is a refinement of the SD hierarchy. Our results will be established with respect to the SMW hierarchy, and then immediately apply to the SD hierarchy.

Our Results and Techniques

We study PMAC-learning the class of monotone, nonnegative, 1-Lipschitz set functions with minimum nonzero value 1 in SMW-$d \supseteq$ SD-d. Parameter d here controls the degree of complementarity in these set functions. In particular, when $d = 0$, we recover the learnability result for submodular functions (Balcan and Harvey 2011).

We restrict our investigation of learnability to product distributions for the following reason: **under arbitrary distributions, every algorithm for PMAC-learning monotone, submodular functions must have approximation factor $\tilde{\Omega}(n^{1/3})$ [1], even if the functions are 1-Lipschitz (Balcan and Harvey 2011).** Note that the maximum possible value of a normalized monotone 1-Lipschitz function is n. In other words, there is no hope for learnability with a decent approximation factor when the underlying distribution is arbitrary. While product distributions may appear not entirely satisfactory in modeling the real world, we argue that the assumption is still to some extent realistic: for example, pawn shops buy items brought to them by different people independently at random, but may sell items in combinations. In general, the assumption holds for any entity that acquires items independently and bundles them for selling.

Breaking the task down. As observed by Balcan and Harvey (Balcan and Harvey 2011), the task of learning submodular set functions can be divided into two parts: learning 0's of the function, and learning the distribution of positive values. We observe a similar phenomenon for functions with complementarity d. We therefore break the task down into two parts, with the first subtask being intrinsically combinatorial, and the second subtask statistical. The plan is to establish learnability for both subtasks respectively, and combine them into learnability of monotone, nonnegative functions with complementarity d.

The combinatorial subtask. In the combinatorial subtask, the goal is to PAC-learn monotone *Boolean-valued* functions in SMW-$d \supseteq$ SD-d. By observing the combinatorial structure of a Boolean-valued SMW-d function, we show that all information of the function is encoded in sets of size not exceeding $d + 1$. This observation immediately leads to an algorithm that PAC-learns these functions using $O(n^{d+1} \log(n/\delta)/\varepsilon)$ samples.

Hardness of learning Boolean-valued functions. We show that, somewhat surprisingly, the combinatorial subtask is the hardcore of learning nonnegative set functions with complementarity d. Specifically, we prove that any algorithm that PAC-learns these functions requires $\tilde{\Omega}(n^{d+1})$ samples, where $\tilde{\Omega}$ hides a polylog factor. Our proof proceeds by constructing a random function f, where the values of f at sets of size smaller than $d + 1$ are 0, and the values at sets of size exactly $d + 1$ are i.i.d., drawn uniformly at random from $\{0, 1\}$. We further fix the product distribution, such that each element $i \in [n]$ appears with probability $(d+1)/n$. We show, that it is hard to learn the values at sets of size $d + 1$ without enough samples. Then, since with constant probability a sample is of size exactly $d + 1$, the algorithm must output a wrong value with constant probability.

The statistical subtask. In the statistical subtask, the goal is to PMAC-learn monotone *positive* functions in SMW-$d \supseteq$ SD-d. We show that, unlike Boolean-valued functions, learning positive functions with constant complementarity with approximation factor $O(1/\varepsilon)$ requires only $O(n^2 \log(1/\delta))$ samples. This bound matches the result for submodular functions up to a constant factor $(d + 1)^2$. The proof proceeds essentially by leveraging the strong concentration properties we establish. Generalizing Balcan and Harvey's result for submodular functions (Balcan and Harvey 2011), we show, that under product distributions, the value of the function converges sharply around the median value, and the mean cannot be too far away from the median. It follows that with high probability, with enough samples, the empirical mean is a good approximation of the value at a random set.

Putting it together. With algorithms for Boolean-valued and positive functions at hand, it is natural to put them together, in the hope that the combination takes care of both subtasks. We show that this is indeed what happens — with approximation factor $O(\log(1/\varepsilon))$, the combination of the two algorithms PMAC-learns nonnegative functions with complementarity d using

$$O(n^2 \log(1/\delta) + n^{d+1} \log(n/\delta)/\varepsilon)$$

samples, where the first term is for positive values, and the second is for 0's. While it may seem weird that samples for learning 0's dominate the total number of samples, we note that the lower bound for learning Boolean-valued functions also applies for nonnegative functions, since the latter subsume the former as a subclass. It follows that the dependency in n in our upper bound is almost tight. We further show, that when the empirical mean is large enough, the number of samples needed becomes significantly smaller. This is because we no longer need to learn the 0's, since concentration bounds guarantee that the probability of a 0 is negligible.

Additional Related Work

Du et al. (Du et al. 2014) and Narasimhan et al. (Narasimhan, Parkes, and Singer 2015) study learning in social networks where the influence function is submodular. Another closely related line of research is on finding succinct approximations for (a.k.a. sketching) complement-free func-

[1] $\tilde{\Omega}$ hides a polylog factor.

tions (Badanidiyuru et al. 2012; Devanur et al. 2013; Cohavi and Dobzinski 2017). There are a number of results on testing submodularity (Seshadhri and Vondrák 2014; Blais and Bommireddi 2017).

Beside the SD and the SMW hierarchies, there are several other measures of complementarity, among which two most useful ones are Maximum-over-Positive-Hypergraphs (MPH) (Feige et al. 2015) and its variant, Maximum-over-Positive-Supermodular (MPS) (Feldman et al. 2016).

The Combinatorial Subtask: Learning Boolean-Valued Functions

In this section we consider PAC-learning of $\mathcal{F}_d^*$, the class of monotone Boolean-valued functions in SMW-$d \supseteq$ SD-d. d can be viewed as a constant measuring the degree of complementarity. As we will see, PAC-learning $\mathcal{F}_d^*$ is the information theoretical hard core of PMAC-learning set functions with limited complementarity.

First we characterize the structure of 0's of a set function in $\mathcal{F}_d^*$. We say $S \subseteq [n]$ is a *zero set* (w.r.t. f) if $f(S) = 0$. Unlike submodular functions, whose system of zero sets is closed downward and under union, a function in $\mathcal{F}_d^*$ may have zero sets with notably more complicated structure. In particular, even if $f(S) = f(T) = 0$, it is not necessarily true that $f(S \cup T) = 0$. To efficiently learn $\mathcal{F}_d^*$, we leverage the following succinct representation of its members' zero sets:

Lemma 1 (Structure of Zero Sets)**.** *For a monotone set function $f \in$ SMW-d, $f(S) = 0$ iff for all $T \subseteq S$ where $|T| \leq d+1$, $f(T) = 0$.*

In other words, all information about f's zero sets is encoded in values of f at sets with size no larger than $d+1$. As a result, to distinguish 0's of f, we only need to keep track of its zero sets of constant size. This characterization leads directly to Algorithm 1.

Proof of Lemma 1. By monotonicity, clearly if $f(S) = 0$ then for any $T \subseteq S$, $f(T) = 0$. We now prove the other direction. Suppose for all $T \subseteq S$ where $|T| \leq d+1$, $f(T) = 0$. We assume $f(S) > 0$ and show a contradiction. Let $S' \subseteq S$ be a subset of S with the smallest cardinality such that $f(S') > 0$. Clearly $|S'| \geq d+2$, and for any $T \subsetneq S'$, $f(T) = 0$. Let v be any element in S'.

$$\begin{aligned} 0 < f(S') &= f(v|S' \setminus \{v\}) \\ &\leq \max\{f(v|T) \mid T \subseteq S' \setminus \{v\},\, |T| \leq d\} \\ &\leq \max\{f(T) \mid T \subseteq S',\, |T| \leq d+1\} \\ &= 0. \end{aligned}$$

□

Theorem 2 (PAC-Learnability of Boolean-Valued Functions)**.** *Restricted to product distributions, for any sufficiently small $\varepsilon > 0$ and $\delta > 0$, Algorithm 1 PAC-learns $\mathcal{F}_d^*$ with parameters (ε, δ). The number of samples is $\ell = 10(d+1)n^{d+1}\log(n/\delta)/\varepsilon$.*

Algorithm 1: An algorithm that PAC-learns monotone Boolean-valued functions with limited complementarity.

Input : ℓ samples $\{(S_i, f^*(S_i))\}_{i\in[\ell]}$ from a product distribution $\mathcal{D}$.
Output: with high probability, an approximately correct estimation f of f^*.

Let $\mathcal{L} \leftarrow \emptyset$.
for $i \in [\ell]$ **do**
  if $f^*(S_i) = 0$*, and there exists a subset T of S_i with $|T| \leq d+1$, such that $T \notin \mathcal{L}$* **then**
    for *Every subset U of S_i with $|U| \leq d+1$* **do**
      Let $\mathcal{L} \leftarrow \mathcal{L} \cup \{U\}$.
Output f, where $f(S) = 0$ iff for any subset T of S with $|T| \leq d+1$, $T \in \mathcal{L}$.

Proof of Theorem 2. We prove a slightly stronger but essentially similar claim, that with probability $1 - \frac{1}{2}\delta$, the algorithm succeeds, and the probability of not recognizing a 0 is at most $\frac{1}{2}\varepsilon$. First note that the family $\mathcal{L}$ contains only sets of size no larger than $d+1$, so its cardinality cannot exceed

$$\sum_{0\leq i\leq d+1} \binom{n}{i} \leq \sum_{0\leq i\leq d+1} n^i = \frac{n^{d+2}-1}{n-1} \leq 2n^{d+1}.$$

Every time we fail to recognize a 0, the size of $\mathcal{L}$ grows by at least 1. So this can happen at most $2n^{d+1}$ times. As long as there is a $\frac{1}{2}\varepsilon$ probability that we fail to recognize a 0, on average we encounter such a sample within $\frac{2}{\varepsilon}$ steps. And in fact, in $8(d+1)\log(n/\delta)/\varepsilon$ steps, the probability of no update is

$$\left(1 - \frac{1}{2}\varepsilon\right)^{8(d+1)\log(n/\delta)/\varepsilon} < \frac{\delta}{4n^{d+1}}.$$

If after some update, the probability that we fail to recognize a 0 drops below $\frac{1}{2}\varepsilon$, the conditions of the theorem are already met. Otherwise, after all updates to $\mathcal{L}$, there is still a $\frac{1}{2}$ probability that we encounter a 0 that cannot be recognized, and the algorithm fails.

We now bound the probability of such a global failure. A union bound over each update immediately gives that after

$$\ell > 10(d+1)n^{d+1}\log(n/\delta)/\varepsilon > 8(d+1)n^{d+1}\log(n/\delta)/\varepsilon$$

steps, the probability that we have not finished the $2n^{d+1}$ updates is at most $\frac{\delta}{2}$. This implies that after seeing all samples, with probability at least $1 - \frac{\delta}{2}$ the algorithm succeeds, in which case the probability that $\mathcal{L}$ fails to recognize a 0 of $f^*(X)$ is at most $\frac{1}{2}\varepsilon$. □

As shown by Balcan and Harvey, PAC-learning Boolean-Valued submodular functions (i.e. $\mathcal{F}_0^*$) under product distributions requires only $O(n\log(n/\delta)/\varepsilon)$ samples. One may question if the n^{d+1} factor for $\mathcal{F}_d^*$ is necessary. We prove the following lower bound, showing that such a factor is necessary for any algorithm that PAC-learns $\mathcal{F}_d^*$.

Theorem 3 (Learning $\mathcal{F}_d^*$ Requires Many Samples)**.** *Fix any $\delta > 0$, $d \in \mathbb{N}$, for large enough n and $\varepsilon < \frac{1}{4}\left(\frac{1}{e}\right)^{d+1}$, any algorithm that PAC-learns $\mathcal{F}_d^*$ with parameters (ε, δ) under a product distribution over $2^{[n]}$ requires $\Omega(n^{d+0.99})$ samples. 0.99 can be replaced by any number smaller than 1.*

Proof sketch. Consider $f^* : 2^{[n]} \to \{0, 1\}$ drawn from the following distribution:

- For S where $|S| \leq d$, $f^*(S) = 0$.
- For S where $|S| = d + 1$, $f^*(S) = 0$ w.p. 0.5. Otherwise $f(S) = 1$.
- For S where $|S| > d + 1$, $f^*(S) = 0$ if for all $T \subseteq S$ where $|T| = d + 1$, $f^*(T) = 0$. Otherwise $f^*(S) = 1$.

It is easy to check that any such function is in SMW-d.

Consider a product distribution over $2^{[n]}$, where each element $i \in [n]$ appears with probability $(d + 1)/n$. One may show that under this distrbution, with constant probability a random set has size precisely $d + 1$. Therefore, to learn f^*, any algorithm must learn correctly the values of most sets of size $d + 1$. There are about n^{d+1} such sets in total.

On the other hand, standard concentration bounds guarantee that a sample set is almost always not too large. In fact, with high probability, all the sample sets have cardinality at most $O(d \log n)$. As a result, only $O((\log n)^{d+1})$ sets of size $d + 1$ can be subsets of a sample set. Conditioned on the event that all sample sets are not too large, since each sample set can only reveal values of f^* at the critical sets it contains, with one sample, the algorithm learns at most $O((\log n)^{d+1})$ values at critical sets. So with relatively few samples, almost always the algorithm fails to learn the values at most critical sets, and therefore does not have enough information about f^*. The lower bound follows. □

The Statistical Subtask: Learning Positive Functions

In this section we consider learning real-valued functions in $\mathcal{F}_d^+$, the family of monotone, positive, 1-Lipschitz set functions with minimum nonzero value 1 in SMW-$d \supseteq$ SD-d. we note that these are standard regularity assumptions for PMAC-learning (Balcan and Harvey 2011).

Concentration with Limited Complementarity

Our most powerful tool for learning $\mathcal{F}_d^+$ is a strong concentration bound, generalizing Balcan and Harvey's result for submodular functions (Balcan and Harvey 2011).

Lemma 2. *Let $f \in$ SMW-d be a monotone, nonnegative, 1-Lipschitz set function with minimum nonzero value 1. Let $\mathcal{D}$ be a product distribution over $2^{[n]}$. For any $b, t \geq 0$,*

$$\Pr_{X\sim\mathcal{D}}\left[f(X) \leq b - t\sqrt{b}\right] \cdot \Pr[f(X) \geq b] \leq \exp(-t^2/4(d+1)^2).$$

The lemma immediately gives concentration around the median. Let $\mathbf{Med}(Z)$ denote the median of real-valued random variable Z. We have:

Corollary 1 (Concentration Around the Median)**.** *Let $f \in$ SMW-d be a monotone, nonnegative, 1-Lipschitz set function with minimum nonzero value 1. Let $\mathcal{D}$ be a product distribution over $2^{[n]}$. For any $t \geq 0$ and $X \sim \mathcal{D}$,*

$$\Pr_{X\sim\mathcal{D}}\left[f(X) - \mathbf{Med}(f(X)) \geq t\sqrt{\mathbf{Med}(f(X))}\right] \leq 2\exp(-t^2/4(d+1)^2), \quad (1)$$

and

$$\Pr_{X\sim\mathcal{D}}\left[\mathbf{Med}(f(X)) - f(X) \geq t\sqrt{\mathbf{Med}(f(X))}\right] \leq 2\exp(-t^2/4(d+1)^2). \quad (2)$$

Proof. Let $b = \mathbf{Med}(X)$ in Lemma 2. We get

$$\begin{aligned}&\Pr\left[f(X) \leq \mathbf{Med}(f(X)) - t\sqrt{\mathbf{Med}(f(X))}\right] \\ &\cdot \Pr[f(X) \geq \mathbf{Med}(f(X))] \\ &\leq \exp(-t^2/4(d+1)^2).\end{aligned}$$

By definition of medians, we have

$$\Pr[f(X) \leq \mathbf{Med}(f(X))] \geq 1/2.$$

So,

$$\begin{aligned}&\Pr\left[f(X) \leq \mathbf{Med}(f(X)) - t\sqrt{\mathbf{Med}(f(X))}\right] \\ &\leq \exp(-t^2/4(d+1)^2)/\Pr[f(X) \leq \mathbf{Med}(f(X))] \\ &\leq 2\exp(-t^2/4(d+1)^2).\end{aligned}$$

Similarly, letting $b = \mathbf{Med}(X) + t\sqrt{\mathbf{Med}(f(X))}$, we get

$$\Pr\left[f(X) \geq \mathbf{Med}(f(X)) + t\sqrt{\mathbf{Med}(f(X))}\right] \leq 2\exp(-t^2/4(d+1)^2).$$

□

We further show that the above concentration bounds around the median can be translated to concentration around the mean, by arguing that the mean is always close to the median:

Lemma 3. *Let $f \in$ SMW-d be a monotone, nonnegative, 1-Lipschitz set function with minimum nonzero value 1. Let $\mathcal{D}$ be a product distribution over $2^{[n]}$. For $X \sim \mathcal{D}$,*

$$\mathbb{E}[f(X)] \geq \frac{1}{2}\mathbf{Med}(f(X)),$$

$$\mathbb{E}[f(X)] \leq \mathbf{Med}(f(X)) + 8(d+1)\sqrt{\mathbf{Med}(f(X))}.$$

Intuitively, these bounds suggest that set functions with limited complementarity, in spite of much weaker separability conditions, have similar concentration behavior to concentration of additive set functions from Hoeffding style arguments.

We note that similar results can be obtained through concentration of self-bounding functions. See, e.g., (Boucheron, Lugosi, and Bousquet 2004; Vondrák 2010). In particular, one may show that every monotone 1-Lipschitz SMW-d function is $(d + 1, 0)$-self-bounding. Concentration of self-bounding functions then yields strong bounds similar to those we present.

Algorithm 2: An algorithm that PMAC-learns monotone positive functions with limited complementarity.

Input : ℓ samples $\{(S_i, f^*(S_i))\}_{i\in[\ell]}$ from a product distribution $\mathcal{D}$.
Output: with high probability, a mostly approximately correct estimation f of f^*.
Let $\mu \leftarrow \frac{1}{\ell}\sum_{i\in[\ell]} f^*(S_i)$.
if $\mu \geq 1000(d+1)^2\log(1/\varepsilon)$ **then**
  Output $f(S) = \frac{\mu}{10}$.
else
  Output $f(S) = 1$.

PMAC-Learning Algorithm for Positive Functions

Equipped with these strong concentration bounds, we are ready to present the PMAC-learning algorithm for $\mathcal{F}_d^+$.

Theorem 4 (PMAC-Learnability of Positive Functions)**.** *Restricted to product distributions, for any sufficiently small $\varepsilon > 0$ and $\delta > 0$, Algorithm 2 PMAC-learns $\mathcal{F}_d^+$ with:*

- *parameters (ε, δ),*
- *approximation factor $\alpha = O((d+1)^2\log(1/\varepsilon))$, and*
- *number of samples $\ell = 10n^2\log(1/\delta)$.*

When $\mathbb{E}[f^*(X)] \geq c(d+1)^2\log(1/\varepsilon)$ for sufficiently large c, the approximation factor improves to 20*.*

Proof sketch. According to Hoeffding's inequality, with high probability the empirical mean μ is an estimation of $\mathbb{E}[f^*(X)]$ with constant additive error. We then proceed by two cases:

1. μ is large enough. This means $\mathbb{E}[f^*(X)]$, and by Lemma 3, $\mathbf{Med}(f^*(X))$, are also large enough. So μ is multiplicatively a good estimation of $\mathbf{Med}(f^*(X))$, and Corollary 1 therefore applies approximately around μ. It is then sufficient to output a constant fraction of μ. The approximation factor α in this case is a constant.
2. μ is relatively small. This means $\mathbb{E}[f^*(X)]$, and by Lemma 3, $\mathbf{Med}(f^*(X))$, are relatively small. It follows from Corollary 1 that with high probability, $f^*(X)$ is close enough to 1. It is then sufficient to output 1, since $f^* \in \mathcal{F}_d^+$ is positive. The approximation factor α in this case is $O((d+1)^2\log(1/\varepsilon))$.

□

Putting Everything Together: Learning General Functions

Now we handle the general case: PMAC-learning $\mathcal{F}_d$, the family of monotone, nonnegative, 1-Lipschitz set functions with minimum nonzero value 1 in SMW-$d \supseteq$ SD-d. With Theorems 2 and 4 at hand, it is natural to combine Algorithms 1 and 2, in the hope of taking care of both the combinatorial and the statistical aspects of the problem simultaneously. We prove, not too surprisingly, that such a combination PMAC-learns $\mathcal{F}_d$.

Algorithm 3: An algorithm that PMAC-learns monotone nonnegative functions with limited complementarity.

Input : ℓ samples $\{(S_i, f^*(S_i))\}_{i\in[\ell]}$ from a product distribution $\mathcal{D}$.
Output: with high probability, a mostly approximately correct estimation f of f^*.
Let $\mu \leftarrow \frac{1}{\ell}\sum_{i\in[\ell]} f^*(S_i)$.
if $\mu \geq 1000(d+1)^2\log(1/\varepsilon)$ **then**
  Output $f(S) = \frac{\mu}{10}$.
else
  Let $\mathcal{L} \leftarrow \emptyset$.
  for $i \in [\ell]$ **do**
    if $f^*(S_i) = 0$*, and there exists a subset T of S_i with $|T| \leq d+1$, such that $T \notin \mathcal{L}$* **then**
      for *Every subset U of S_i with $|U| \leq d+1$* **do**
        Let $\mathcal{L} \leftarrow \mathcal{L} \cup \{U\}$.
  Output f, where $f(S) = 0$ if for any subset T of S with $|T| \leq d+1$, $T \in \mathcal{L}$, and $f(S) = 1$ otherwise.

Theorem 5 (PMAC-Learnability of General Functions)**.** *Restricted to product distributions, for sufficiently small $\varepsilon > 0$ and $\delta > 0$, Algorithm 3 PMAC-learns $\mathcal{F}_d$ with:*

- *parameters (ε, δ),*
- *approximation factor $\alpha = O((d+1)^2\log(1/\varepsilon))$, and*
- *number of samples $\ell = 10n^2\log(1/\delta) + 10(d+1)n^{d+1}\log(n/\delta)/\varepsilon$.*

When $\mathbb{E}[f^*(X)] \geq c(d+1)^2\log(1/\varepsilon)$ for sufficiently large c, the approximation factor improves to 20*, and the number of samples improves to $\ell = 10n^2\log(1/\delta)$.*

Proof sketch. Again, consider the two cases:

1. μ is large enough. Lemma 3 and Corollary 1 establish the same strong concentration around μ, so it is sufficient to output a constant fraction of μ. In this case we do not need samples in the next phase, so the number of samples needed is $10n^2\log(1/\delta)$. And for similar reasons, the approximation factor is 20.
2. μ is relatively small. Because f^* is now nonnegative, we need to consider 0's of f^*. When our estimation is wrong, it can be either (1) $f(S) = 1$ and $f^*(S) = 0$, or (2) $f(S) = 1$ and $f^*(S)$ is too large. By adapting Theorem 2 we can show that situation (1) happens with probability at most $\frac{1}{2}\varepsilon$, and by Theorem 4, situation (2) happens with probability at most $\frac{1}{2}\varepsilon$. The overall probability of a wrong estimation is at most ε. In this case we need all $\ell = 10n^2\log(1/\delta) + 10(d+1)n^{d+1}\log(n/\delta)/\varepsilon$ samples, and the approximation factor is $O((d+1)^2\log(1/\varepsilon))$.

□

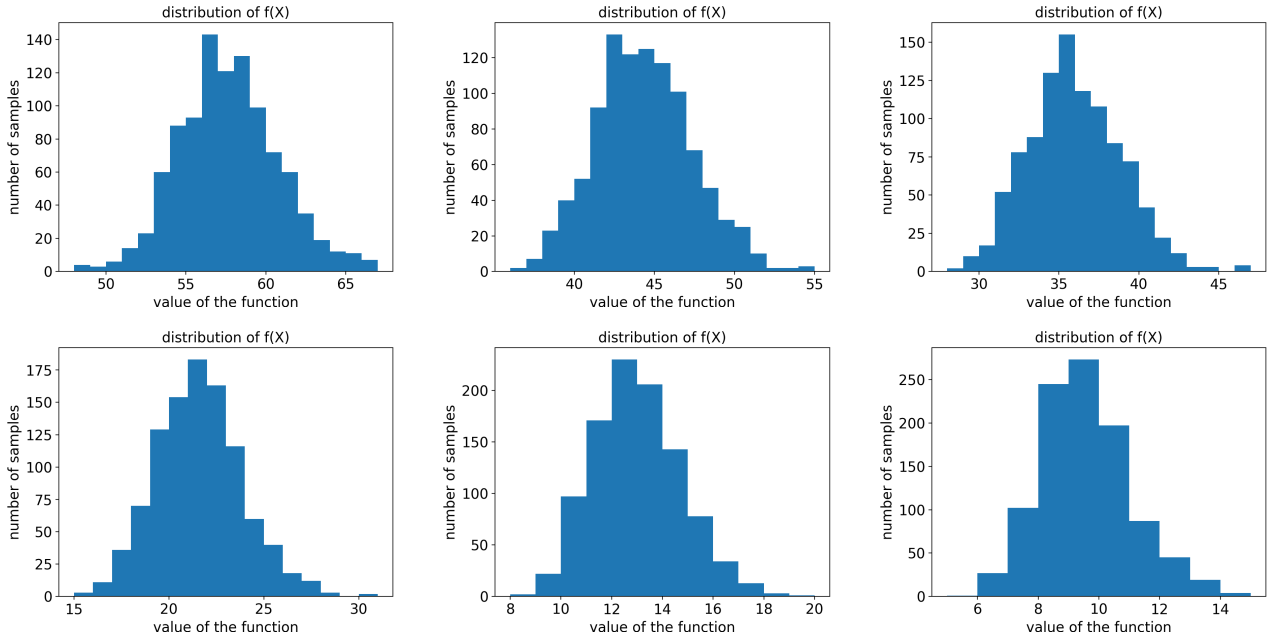

Figure 1: Distributions of $f(X)$ when the degree of complementarity is 0, 1, 2, 5, 10, and 15.

Note that since $\mathcal{F}_d$ subsumes $\mathcal{F}_d^*$, Theorem 3 directly implies the same information theoretical lower bound for PMAC-learning $\mathcal{F}_d$, complementing Theorem 5. Formally:

Corollary 2 (Learning $\mathcal{F}_d$ Requires Many Samples)**.** *Fix any $\delta > 0$, $d \in \mathbb{N}$, for large enough n and $\varepsilon < \frac{1}{4}\left(\frac{1}{e}\right)^{d+1}$, any algorithm that PAC-learns $\mathcal{F}_d$ with parameters (ε, δ) under a product distribution over $2^{[n]}$, with any approximation factor, requires $\omega(n^{d+0.99})$ samples. 0.99 can be replaced by any number smaller than 1.*

We note again, that since the SMW hierarchy is strictly more expressive than the SD hierarchy (Proposition 1), all our learnability results for functions in $\mathcal{F}_d^*$, $\mathcal{F}_d^+$ and $\mathcal{F}_d$ apply to functions with the same set of requirements but with SMW-d replaced by SD-d.

Numerical Simulations

We conduct numerical simulations to investigate the empirical concentration of set functions with limited complementarity. Unfortunately, there is no efficient way known to generate set functions in SMW-d or SD-d. Instead, we sample functions from the Maximum-over-Positive-Hypergraphs (MPH) hierarchy (Feige et al. 2015), which is another well-known measure of complementarity. It has also been used to study the equilibrium behavior of agents in certain auction protocols (Feige et al. 2015). In some sense, the experimental results with MPH functions should be considered a complement to our theoretic results, as it sheds light on the statistical behavior of functions considered to have limited complementarity according to another commonly used measure, even if Theorem 5 does not always provide strong guarantees for them.

The MPH hierarchy builds on the concept of *hypergraphs*. Roughly speaking, a hypergraph can be viewed as a set function, where the value of a set is the sum of the weights of hyperedges that the set contains. A function is in MPH-$(d+1)$, if there exists a finite number of hypergraphs containing only positively weighted hyperedges of size at most $(d+1)$, such that the value of the function at any set is the maximum value of the set in these hypergraphs. MPH-0 is exactly the class of functions that are fractionally subadditive.

In our experiments, we fix the cardinality of the ground set to be $n = 1000$ and the number of hypergraphs to be 10. In each hypergraph, we choose uniformly at random 100 disjoint sets of size chosen from $\{1, 2, \ldots, d+1\}$ uniformly at random. For each degree of complementarity d, we sample a function f in MPH-d in the way described above, draw 1000 sample sets where each element appears with probability 0.5, and plot the empirical distribution of $f(X)$.

As can be seen from Figure 1, the degradation of concentration is remarkably smooth as d grows from 0 to 15. Even when $d = 15$, most samples still lie between 6 and 14, where the multiplicative gap is only about 2.33. The experimental results suggest that set functions in the real world are likely to exhibit decent concentration, since it is intuitively unlikely that more than 15 items are involved, altogether as complements to each other. With strong concentration like this, anything close to the (empirical) mean of the function value at a random set is a good approximation of the value at any set. In such cases, it is reasonable to believe that our learning algorithm works well with less samples and better approximation ratios than what the theory guarantees.

Acknowledgements

We are thankful for support from NSF under awards IIS-1814056 and IIS-1527434. We also thank Wei Chen, Shang-Hua Teng, and anonymous reviewers for helpful feedback.

References

Bach, F., et al. 2013. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning* 6(2-3):145–373.

Badanidiyuru, A.; Dobzinski, S.; Fu, H.; Kleinberg, R.; Nisan, N.; and Roughgarden, T. 2012. Sketching valuation functions. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, 1025–1035. Society for Industrial and Applied Mathematics.

Balcan, M.-F., and Harvey, N. J. 2011. Learning submodular functions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, 793–802. ACM.

Balcan, M. F.; Constantin, F.; Iwata, S.; and Wang, L. 2012. Learning valuation functions. In *Conference on Learning Theory*, 4–1.

Blais, E., and Bommireddi, A. 2017. Testing submodularity and other properties of valuation functions. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 67. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Boucheron, S.; Lugosi, G.; and Bousquet, O. 2004. Concentration inequalities. In *Advanced Lectures on Machine Learning*. Springer. 208–240.

Chen, W.; Teng, S.-H.; and Zhang, H. 2019. Capturing Complementarity in Set Functions by Going Beyond Submodularity/Subadditivity. In *Proceedings of the 10th conference on Innovations in Theoretical Computer Science*.

Cohavi, K., and Dobzinski, S. 2017. Faster and simpler sketches of valuation functions. *ACM Transactions on Algorithms (TALG)* 13(3):30.

Devanur, N. R.; Dughmi, S.; Schwartz, R.; Sharma, A.; and Singh, M. 2013. On the approximation of submodular functions. *arXiv preprint arXiv:1304.4948*.

Du, N.; Liang, Y.; Balcan, M.; and Song, L. 2014. Influence function learning in information diffusion networks. In *International Conference on Machine Learning*, 2016–2024.

Eden, A.; Feldman, M.; Friedler, O.; Talgam-Cohen, I.; and Weinberg, S. M. 2017. A simple and approximately optimal mechanism for a buyer with complements. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, 323–323. ACM.

Feige, U., and Izsak, R. 2013. Welfare maximization and the supermodular degree. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, 247–256. ACM.

Feige, U.; Feldman, M.; Immorlica, N.; Izsak, R.; Lucier, B.; and Syrgkanis, V. 2015. A unifying hierarchy of valuations with complements and substitutes. In *AAAI*, 872–878.

Feige, U. 2009. On maximizing welfare when utility functions are subadditive. *SIAM Journal on Computing* 39(1):122–142.

Feldman, M., and Izsak, R. 2014. Constrained monotone function maximization and the supermodular degree. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques* 160.

Feldman, M.; Friedler, O.; Morgenstern, J.; and Reiner, G. 2016. Simple mechanisms for agents with complements. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, 251–267. ACM.

Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 137–146. ACM.

Lin, H., and Bilmes, J. 2012. Learning mixtures of submodular shells with application to document summarization. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 479–490. AUAI Press.

Narasimhan, M., and Bilmes, J. A. 2007. Local search for balanced submodular clusterings. In *IJCAI*, 981–986.

Narasimhan, H.; Parkes, D. C.; and Singer, Y. 2015. Learnability of influence in networks. In *Advances in Neural Information Processing Systems*, 3186–3194.

Nemhauser, G. L., and Wolsey, L. A. 1978. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research* 3(3):177–188.

Seshadhri, C., and Vondrák, J. 2014. Is submodularity testable? *Algorithmica* 69(1):1–25.

Vondrák, J. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, 67–74. ACM.

Vondrák, J. 2010. A note on concentration of submodular functions. *arXiv preprint arXiv:1005.2791*.

Omitted Proofs

Proof of Theorem 3. Consider $f^* : 2^{[n]} \to \{0, 1\}$ drawn from the following distribution:

- For S where $|S| \le d$, $f^*(S) = 0$.
- For S where $|S| = d + 1$, $f^*(S) = 0$ w.p. 0.5. Otherwise $f(S) = 1$.
- For S where $|S| > d + 1$, $f^*(S) = 0$ if for all $T \subseteq S$ where $|T| = d + 1$, $f^*(T) = 0$. Otherwise $f^*(S) = 1$.

It is easy to check that any such function is in SMW-d.

To simplify notations let $D = d+1$. Consider a distribution $\mathcal{D}$ over $2^{[n]}$, where each element $i \in [n]$ appears with probability D/n. Fix a possibly randomized learning algorithm. Suppose, after seeing $\ell = O(n^{d+0.99})$ samples, the algorithm correctly outputs $f^*(S)$ for $S \sim \mathcal{D}$ with probability $1 - \varepsilon$. To make the task easier we further assume that (1) the algorithm is aware of the distribution from which f is drawn, and (2) a sample S reveals $f^*(T)$ for all $T \subseteq S$. According to our assumption, the task is essentially to learn $f^*(T)$ for all set T where $|T| = D$, which can be accomplished only by observing sample sets with size no smaller than D. We call these sets of size exactly D *critical*.

First we show that with high probability, all samples that the algorithm sees are of reasonably small size. Consider a binomial distribution with parameters n and D/n, denoted by $\mathcal{B}$. Let $X \sim \mathcal{B}$. By definition we know that for any $i > 0$,

$$\frac{\Pr[X = i]}{\Pr[X = i - 1]} = \frac{(n - i - 1)D/n}{(i + 2)(1 - D/n)} \le \frac{2D}{i}.$$

For $i \ge 4D$, $\frac{\Pr[X=i+1]}{\Pr[X=i]} \le \frac{1}{2}$. Therefore for $i \ge 4D$,

$$\Pr[X = i] \le \left(\frac{1}{2}\right)^{i-4D} \Pr[X = 4D] \le \left(\frac{1}{2}\right)^{i-4D}.$$

In particular, for $i \ge 3D \log n$,

$$\Pr[X = i] \le n^{-2D}.$$

Now let $S_1, \ldots, S_l$ be the samples the algorithm sees. Clearly $|S_i| \sim \mathcal{B}$, so

$$\Pr[\wedge_i(|S_i| \le 3D \log n)] \ge 1 - \sum_{i \in [l]} \Pr[|S_i| \ge 3D \log n] \ge 1 - \ell n^{-2D} = 1 - o(1).$$

In other words, with probability $1 - o(1)$ all sample sets are of size no larger than $3D \log n$.

Now we consider the number of critical sets whose values are not revealed by the ℓ samples. In particular, we show this number is large with high probability. Conditioned on $|S_i| \le 3D \log n$, the number of critical sets which are subsets of S_i is

$$\binom{3D \log n}{D} \le \left(\frac{10D \log n}{D}\right)^D = (10 \log n)^D.$$

So with probability $1 - o(1)$, each sample reveals the values of no more than $(10 \log n)^D$ critical sets, and after seeing all ℓ samples, there are still at least

$$\binom{n}{D} - \ell(10 \log n)^D = \binom{n}{D} - (10 \log n)^D \cdot O(n^{D-0.01}) \ge \frac{1}{2}\binom{n}{D}.$$

That is, with high probability, half of the critical sets are still not touched by any sample. If any of these sets are queried, the best the algorithm can do is random guess, with a success probability of 0.5.

Now we consider the probability that such a set is drawn from distribution $\mathcal{D}$. If the probability is larger than 2ε, the algorithm will give a wrong answer with probability larger than ε, which leads to a contradiction. First consider the probability that $S \sim \mathcal{D}$ is of size exactly D.

$$\begin{aligned}
\Pr[|S| = D] &= \binom{n}{D}\left(\frac{D}{n}\right)^D\left(1 - \frac{D}{n}\right)^{n-D} \\
&\ge \left(\frac{n}{D}\right)^D\left(\frac{D}{n}\right)^D\left(\left(1 - \frac{D}{n}\right)^{n/D}\right)^D\left(1 - \frac{D}{n}\right)^{-D} \\
&\ge \left(\frac{1}{e}\right)^D.
\end{aligned}$$

Also note that

$$\Pr[S \text{ is not touched by any sample} \mid |S| = D] \geq \frac{1}{2}\binom{n}{D}/\binom{n}{D} = \frac{1}{2}.$$

So for any $\varepsilon < \frac{1}{4}\left(\frac{1}{e}\right)^D$, with probability $1 - o(1)$, the probability that the algorithm gives a wrong answer on a critical set that is not touched by any sample is already larger than ε. □

Proof of Lemma 2. We prove an equivalent form: For any $b, t \geq 0$,

$$\Pr\left[f(X) \leq b - t(d+1)\sqrt{b}\right] \cdot \Pr[f(X) \geq b] \leq \exp(-t^2/4).$$

Assume $t(d+1) \leq \sqrt{b}$, since otherwise $\Pr\left[f(X) \leq b - t(d+1)\sqrt{b}\right] = 0$. Talagrand's inequality states that for $\mathcal{A} \subseteq \{0,1\}^n$, and $y \in [0,1]^n$ drawn from a product distribution,

$$\Pr[y \in \mathcal{A}] \cdot \Pr[\rho(\mathcal{A}, y) < t] \leq \exp(-t^2/4),$$

where ρ is a distance function defined by

$$\rho(\mathcal{A}, y) = \sup_{\alpha \in \mathbb{R}^n, \|\alpha\|_2 = 1} \min_{z \in \mathcal{A}} \sum_{i : y_i \neq z_i} \alpha_i.$$

We apply this to $\mathcal{A} = \left\{X : f(X) < b - t(d+1)\sqrt{b}\right\}$.

We show that for any Y, $f(Y) \geq b$ implies $\rho(\mathcal{A}, Y) > t$. If this is true, then $\Pr[f(Y) \geq b] \leq \Pr[\rho(\mathcal{A}, Y) > t]$ and the lemma follows. Suppose $\rho(\mathcal{A}, Y) \leq t$. W.l.o.g. we relabel Y to be $Y = \{1, \ldots, k\} = [k]$, such that for any $i \in (d+1)\mathbb{N} \cap Y = \{0, d+1, 2(d+1), \ldots\} \cap Y$.

$$f(\{i+1, \ldots, i+d+1\} \cap Y | [i]) = \max\{f(S|[i]) \mid S \subseteq Y \setminus [i], |S| = \min(k-i, d+1)\}.$$

That is, $\{i+1, \ldots, i+d+1\} \cap Y$ achieves the maximum marginal given $[i]$. Let $E_i = [i(d+1)] \cap Y$, $\ell(i) = \max\{(d+1)\mathbb{N} \cap [i-1]\}$, $r(i) = \min\{((d+1)\mathbb{N} \cup \{k\}) \setminus [i-1]\}$. Note that with these definitions, the property from relabeling of Y translates to

$$\forall i,\ f(\{i(d+1)+1, \ldots, i(d+2)\} \cap Y | E_i) = \max\{f(S|E_i) \mid S \subseteq Y \setminus E_i, |S| = \min(k-i, d+1)\}.$$

Define

$$\alpha_i = \begin{cases} f\left(E_{r(i)}\right) - f\left(E_{\ell(i)}\right), & \text{if } i \in Y, \\ 0, & \text{otherwise.} \end{cases}.$$

Since f is monotone and 1-Lipschitz, $0 \leq \alpha_i \leq d+1$, and so

$$\|\alpha\|_2 \leq \sqrt{(d+1)\sum_i \alpha_i} \leq (d+1)\sqrt{f(Y)}.$$

Since we suppose $\rho(\mathcal{A}, Y) > t$, the definition of ρ implies that there is some $Z \in \mathcal{A}$, such that

$$\sum_{(Y \setminus Z) \cup (Z \setminus Y)} \alpha_i \leq \rho(\mathcal{A}, Y)\|\alpha_2\| \leq t(d+1)\sqrt{f(Y)}.$$

From monotonicity of f and the fact that $\mathcal{A}$ is downward closed, we may assume w.l.o.g. $Z \subseteq Y$, since $Z \cap Y$ also satisfies the above conditions.

Let $V = Y \setminus Z = \{v_1, \ldots, v_m\}$ where $v_i < v_j$ for $i < j$, and $F_i = \{v_1, \ldots, v_i\} \cup Z$ for $i \in [m]$. Now we show a

contradiction by deriving $f(Y) - f(Z) \le t(d+1)\sqrt{f(Y)}$.

$$\begin{aligned}
f(Y) - f(Z) &= \sum_{i\in[m]} f(F_i) - f(F_{i-1}) \\
&= \sum_{i\in[m]} f(v_i|F_{i-1}) \\
&= \sum_{i\in[m]} f(v_i|(F_{i-1}\cap E_{\ell(v_i)}) \cup (F_{i-1}\cap([k]\setminus E_{\ell(v_i)}))) \\
&= \sum_{i\in[m]} f(v_i|E_{\ell(v_i)} \cup (F_{i-1}\cap([k]\setminus E_{\ell(v_i)}))) \\
&\le \sum_{i\in[m]} f(v_i|E_{\ell(v_i)}\cup S_i), \text{ where } S_i \subseteq [k]\setminus E_{\ell(v_i)},\ |S_i| \le d \\
&\le \sum_{i\in[m]} f(\{v_i\}\cup S_i|E_{\ell(v_i)}), \text{ where } (S_i\cup\{v_i\} \subseteq [k]\setminus E_{\ell(v_i)},\ |S_i\cup\{v_i\}| \le d+1 \\
&\le \sum_{i\in[m]} f(\{\ell(v_i)+1,\dots,\ell(v_i)+d+1\}|E_{\ell(v_i)}) \\
&= \sum_{i\in[m]} \alpha_{v_i} \\
&= \sum_{i\in Y\setminus Z} \alpha_i \\
&\le t(d+1)\sqrt{f(Y)}.
\end{aligned}$$

So $f(Z) \ge f(Y) - t(d+1)\sqrt{f(Y)} \ge b - t(d+1)\sqrt{b}$ since $f(Y) \ge b$ and $t(d+1) \le \sqrt{b}$. This contradicts $Z \in \mathcal{A}$. □

Proof of Lemma 3. First note that by Markov bound,

$$\frac{1}{2} \le \Pr[f(X) \ge \mathbf{Med}(f(X))] \le \frac{\mathbb{E}[f(X)]}{\mathbf{Med}(f(X))}.$$

The left inequality follows immediately.

For the right inequality, consider an estimation based on (1).

$$\begin{aligned}
\mathbb{E}[f(X)] &\le \mathbf{Med}(f(X)) + 2\sqrt{\mathbf{Med}(f(X))}\int_0^\infty \exp(-t^2/4(d+1)^2)\mathrm{d}t \\
&\le \mathbf{Med}(f(X)) + 2\sqrt{\mathbf{Med}(f(X))}\int_0^{2(d+1)} \exp(-t^2/4(d+1)^2)\mathrm{d}t \\
&+ 2\sqrt{\mathbf{Med}(f(X))}\int_{2(d+1)}^\infty \exp(-t^2/4(d+1)^2)\mathrm{d}t \\
&\le \mathbf{Med}(f(X)) + 4(d+1)\sqrt{\mathbf{Med}(f(X))} \\
&+ 2\sqrt{\mathbf{Med}(f(X))}\int_{2(d+1)}^\infty \exp(-t^2/4(d+1)^2)\mathrm{d}t \\
&\le \mathbf{Med}(f(X)) + 4(d+1)\sqrt{\mathbf{Med}(f(X))} \\
&+ 2\sqrt{\mathbf{Med}(f(X))}\int_{2(d+1)}^\infty \exp(-t/2(d+1))\mathrm{d}t \\
&\le \mathbf{Med}(f(X)) + 4(d+1)\sqrt{\mathbf{Med}(f(X))} + 4(d+1)\sqrt{\mathbf{Med}(f(X))} \\
&\le \mathbf{Med}(f(X)) + 8(d+1)\sqrt{\mathbf{Med}(f(X))}.
\end{aligned}$$

□

Lemma 4. *For ε small enough, if $\mu \ge 1000(d+1)^2\log(1/\varepsilon)$, then with probability $1-\delta$ Algorithm 2 outputs a 20-approximation of $f^*(X)$ w.p. $1-\varepsilon$.*

Proof. By a Hoeffding style argument,

$$\Pr[|\mu - \mathbb{E}[f^*(X)]| > 1] < 2\exp(-2\ell/n^2) \leq 2\exp(-20\log(1/\delta)) < \delta.$$

So with probability at least $1-\delta$, $\mathbb{E}[f^*(X)] > \mu - 1 \geq 900(d+1)^2\log(1/\varepsilon)$. By Lemma 3,

$$\mathbf{Med}(f^*(X)) \geq 500(d+1)^2\log(1/\varepsilon) \geq \frac{1}{2}\mathbb{E}[f^*(X)].$$

Now consider the probability that $1/10\mu$ is a 20-approximation of $f^*(X)$ (i.e. $f^*(X)$ is in $[1/10\mu, 2\mu]$). By relationship of μ, $\mathbb{E}[f^*(X)]$ and $\mathbf{Med}(f^*(X))$ derived above and Corollary 1,

$$\begin{aligned}
&\Pr[1/10\mu \leq f^*(X) \leq 2\mu] \\
\geq\ &\Pr\left[\frac{1}{2}\mathbf{Med}(f^*(X)) \leq f^*(X) \leq \frac{3}{2}\mathbf{Med}(f^*(X))\right] \\
= 1 - &\Pr\left[f^*(X) - \mathbf{Med}(f^*(X)) > \frac{1}{2}\sqrt{\mathbf{Med}(f^*(X))}\sqrt{\mathbf{Med}(f^*(X))}\right] \\
- &\Pr\left[\mathbf{Med}(f^*(X)) - f^*(X) > \frac{1}{2}\sqrt{\mathbf{Med}(f^*(X))}\sqrt{\mathbf{Med}(f^*(X))}\right] \\
&\geq 1 - 4\exp\left(-\mathbf{Med}(f^*(X))/16(d+1)^2\right) \\
&\geq 1 - 4\exp(-10\log(1/\varepsilon)) \\
&\geq 1 - \varepsilon.
\end{aligned}$$

□

Lemma 5. *For ε small enough, if $\mu < 1000(d+1)^2\log(1/\varepsilon)$, then conditioned on $f^*(X) > 0$, with probability $1 - \frac{1}{2}\delta$ Algorithm 2 outputs a $3000(d+1)^2\log(1/\varepsilon)$-approximation of $f^*(X)$ w.p. $1 - \frac{1}{2}\varepsilon$.*

Proof. Recall that with probability $1-\delta$,

$$|\mu - \mathbb{E}[f^*(X)]| \leq 1.$$

So

$$\mathbf{Med}(f^*(X)) \leq 2\mathbb{E}[f^*(X)] \leq 2200(d+1)^2\log(1/\varepsilon).$$

Since $f^*(X) > 0$, the only possibility that $f^*(X)$ is not approximated by 1 is $f^*(X)$ being too large. We now bound this probability. By (1) in Corollary 1,

$$\begin{aligned}
&\Pr[f^*(X) > 3000(d+1)^2\log(1/\varepsilon)] \\
\leq\ &\Pr\left[f^*(X) - \mathbf{Med}(f^*(X)) > 10(d+1)\sqrt{\log(1/\varepsilon)}\sqrt{\mathbf{Med}(f^*(X))}\right] \\
&\leq 2\exp(-100(d+1)^2\log(1/\varepsilon)/16(d+1)^2) \\
&\leq \frac{1}{2}\varepsilon.
\end{aligned}$$

□

Proof of Theorem 4. When $\mu \geq 1000(d+1)^2\log(1/\varepsilon)$, Lemma 4 guarantees 20-approximation with parameters (ε, δ). When $\mu < 1000(d+1)^2\log(1/\varepsilon)$, Lemma 5 guarantees $3000(d+1)^2\log(1/\varepsilon)$-approximation with parameters $(1/2\varepsilon, \delta)$. Together Algorithm 2 PMAC-learns $\mathcal{F}_d^+$ with parameters (ε, δ) and approximation factor $O((d+1)^2\log(1/\varepsilon))$. □

Proof of Theorem 5. When $\mu \geq 1000(d+1)^2\log(1/\varepsilon)$, by Lemma 4 the statement holds. When $\mu < 1000(d+1)^2\log(1/\varepsilon)$, with probability $1-\delta$, both Lemma 5 and Theorem 2 work. In that case, with probability $\frac{1}{2}\varepsilon$ we fail to approximate a positive $f^*(X)$, and with probability $\frac{1}{2}\varepsilon$ (as we show in the proof of Theorem 2, instead of ε) we fail to recognize a 0. The total probability of a successful approximation is at least $1-\varepsilon$. □