Learning Submodular Functions

Maria-Florina Balcan* Nicholas J. A. Harvey†

Abstract

This paper considers the problem of learning submodular functions. A problem instance consists of a distribution on $\{0,1\}^n$ and a real-valued function on $\{0,1\}^n$ that is non-negative, monotone and submodular. We are given $\mathrm{poly}(n)$ samples from this distribution, along with the values of the function at those sample points. The task is to approximate the value of the function to within a multiplicative factor at subsequent sample points drawn from the distribution, with sufficiently high probability. We prove several results for this problem.

- There is an algorithm that, for any distribution, approximates any such function to within a factor of $\sqrt{n}$ on a set of measure $1-\epsilon$.
- There is a distribution and a family of functions such that no algorithm can approximate those functions to within a factor of $\tilde{O}(n^{1/3})$ on a set of measure $1/2+\epsilon$.
- If the function is a matroid rank function and the distribution is a product distribution then there is an algorithm that approximates it to within a factor $O(\log(1/\epsilon))$ on a set of measure $1-\epsilon$.

Our study involves a twist on the usual learning theory models and uncovers some interesting extremal properties of submodular functions.

1 Introduction

Submodular functions are a discrete analog of convex functions that enjoy numerous applications and have structural properties that can be exploited algorithmically. They arise naturally in the study of graphs, matroids, covering problems, facility location problems, etc., and they have been extensively studied in operations research and combinatorial optimization for many years [16]. More recently submodular functions have become key concepts both in the machine learning and algorithmic game theory communities. For example, submodular functions have been used to model bidders' valuation functions in combinatorial auctions [28, 8, 4, 37], and for solving feature selection problems in graphical models [25] or for solving various clustering problems [30]. In fact, submodularity in machine learning has been a topic of two tutorials at two recent major conferences in machine learning [26, 27].

Despite the increased interest on submodularity in machine learning, little is known about the topic from a learning theory perspective. One exception is the recent work of Goemans et al. [15], who considered learning submodular functions with value queries in an exact learning model. In this work, we consider submodular function learning in the more traditional distributional learning setting, and we develop a PAC-style analysis for this scenario.

*Georgia Institute of Technology, School of Computer Science. Email: ninamf@cc.gatech.edu. Portions of this work were done while at Microsoft Research, New England.

†University of Waterloo, Department of Combinatorics and Optimization. Email: harvey@math.uwaterloo.ca. Portions of this work were done while at Microsoft Research, New England.

The model we consider in this paper is as follows. The learning algorithm is given a set $\mathcal{S}$ of polynomially many labeled examples drawn i.i.d. from some fixed, but unknown, distribution D over points in $\{0,1\}^n$; alternatively one may think of D as a distribution over subsets of $\{1,\ldots,n\}$. The points are labeled by a fixed, but unknown, target function f^*. The function f^* is assumed to be real-valued, non-negative, monotone and submodular.[1] The goal is to output a hypothesis function f that, with probability $1-\delta$ over the choice of examples, is a good approximation of the target f^* on most of the points coming from D. Here "most" means a $1-\epsilon$ fraction and "good approximation" means that $f(S) \leq f^*(S) \leq g(n) \cdot f(S)$ for some function $g(n)$. We prove that that it if the underlying distribution is arbitrary and the target function is an arbitrary non-negative, monotone, submodular function, then it is not possible to achieve a constant approximation factor $g(n)$. Given this, for most of the paper we focus on the question: What approximation factors $g : \mathbb{N} \to \mathbb{R}$ are possible? We prove $\mathrm{poly}(n)$ upper and lower bounds on the approximation factor achievable when the algorithm receives only $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$ examples.

Our learning model described above differs from the usual PAC-learning model; we call our model the PMAC-learning model. (The "M" stands for "mostly".) In our model, one must *approximate* the value of a function on a set of large measure, with high confidence. In contrast, the traditional PAC-learning model usually studies learnability of much simpler classes of Boolean functions. There, one must compute the value *exactly* on a set of large measure, with high confidence.

Our study has multiple motivations. From an applications perspective, algorithms for learning submodular functions may be useful in some of the applications where these functions arise. For example, an auctioneer may use such an algorithm to "sketch" the players' valuation functions before designing the auction. (We ignore the players' incentives in this example.) From a foundational perspective, submodular functions are a powerful, broad class of important functions, so studying their learnability allows us to understand their structure in a new way. To draw a parallel to the Boolean-valued case, a class of comparable breadth would be the class of monotone Boolean functions; the learnability of such functions has been intensively studied [5, 6].

Our work has several interesting by-products. One is the PMAC-learning model, which studies both the probability mass of points on which the hypothesis does well and the multiplicative approximation achieved on those points. Another by-product of our work is a new family of matroids which reveals interesting extremal properties of submodular functions. Roughly speaking, we show that a small Boolean cube can be embedded into a large Boolean cube so that *any* $\{0,1\}$-valued function on the small cube maps to a function that is submodular on the large cube but is now $\{\alpha,\beta\}$-valued (on the image of the embedding) with $\alpha \ll \beta$.

1.1 Overview of Our Results and Techniques

When studying learning of submodular functions, a natural starting point would be to consider Boolean-valued submodular functions. However, these functions are such a restricted class that learning them is trivial, as we show in Appendix E. For the remainder of this paper we focus on real-valued submodular functions.

We start by showing that it is possible to PMAC-learn the general class of non-negative, monotone submodular functions with an approximation factor of $\sqrt{n+1}$. To prove this we use a structural result in [15] which shows that any monotone, non-negative submodular function can be approximated within a factor of $\sqrt{n}$ on every point by the square root of an additive function. Using this result, we show how to convert the problem of learning a submodular function in the PMAC model to the problem of learning a linear sep-

[1] Formally, $f^*(S) \geq 0$ for all $S \subseteq \{1,\ldots,n\}$, $f^*(S) \leq f^*(T)$ whenever $S \subseteq T \subseteq \{1,\ldots,n\}$, and $f^*(S) + f^*(T) \geq f^*(S \cup T) + f^*(S \cap T)$ for all $S, T \subseteq \{1,\ldots,n\}$.

arator in $\mathbb{R}^{n+1}$ in the usual PAC model. We remark that an improved structural result for any subclass of submodular functions immediately implies an improved analysis of our algorithm for that subclass.

We use the new matroid family mentioned above to show a comparable lower bound: any algorithm that uses a polynomial number of examples cannot PMAC-learn the class of submodular functions with an approximation factor $o(n^{1/3}/\log n)$. In fact, we show that even *weak* PMAC-learning is not possible — any algorithm can do only negligibly better than random guessing for this class of functions. Moreover, this lower bound holds even if the algorithm is told the underlying distribution and it is given the ability to query the function on inputs of its choice and even if the queries are adaptive. In other words this lower bound holds even in the PMAC model augmented with value queries.

This lower bound holds even for matroid rank functions, but it uses a distribution on inputs which is a non-product distribution. It turns out that the use of such a distribution is necessary: using Talagrand's inequality, we prove that a constant approximation factor can be achieved for matroid rank functions under product distributions.

To prove the lower bound, we consider the following technical problem. We would like find an injective map $\rho : \{0,1\}^d \to \{0,1\}^n$ and real numbers $\alpha \gg \beta$ such that every Boolean function f on $\{0,1\}^d$ can be mapped to a non-negative, monotone, submodular function $\tilde{f}$ on $\{0,1\}^n$ satisfying $f(x) = 0 \Rightarrow \tilde{f}(\rho(x)) \leq \beta$ and $f(x) = 1 \Rightarrow \tilde{f}(\rho(x)) \geq \alpha$. This implies a lower bound on learning submodular functions with approximation factor $\frac{\alpha}{\beta}$ when $d = \omega(\log n)$. A trivial construction is obtained using partition matroids, with $\beta = 0$, $\alpha \leq \frac{n}{2}$ and $d \leq \log(\lfloor n/\alpha \rfloor)$; here d is too small to be of interest. Another easy construction is obtained using paving matroids, with $\beta = \frac{n}{2}$, $\alpha = \frac{n}{2} + 1$, and any $d = n - \Omega(\log^4(n))$; here d is large, but there is only a small additive gap between α and β. Our new family of matroids is a common generalization of partition and paving matroids. We use them to obtain a construction with $\beta = 16d$, $\alpha = n^{1/3}$ and any $d = o(n^{1/3})$. Setting $d = \log n \log\log n$ gives the $\tilde{\Omega}(n^{1/3})$ lower bound for learning submodular functions.

1.2 Related Work

Exact Learning with Value Queries. As mentioned above, a recent paper of Goemans et al. [15] also considers the problem of approximately learning non-negative, monotone, submodular functions. From a learning theory perspective, the results of Goemans et al. [15] are of the type "exact learning in the value query model". In particular, their model does not involve a distribution on the domain of the function. Instead, the algorithm queries the domain at $\mathrm{poly}(n)$ points (chosen adaptively by the algorithm) and it learns the value of f^* at those points. Then the algorithm must produce a hypothesis f which is correct to within a multiplicative factor $g(n)$ for *every* point in the domain. In contrast, the model that we focus on is the more widely studied *passive supervised learning* setting [3, 24, 35, 36]. The algorithm can only see a polynomial number of examples chosen i.i.d. from the underlying distribution, but only requires that the algorithm approximate f^* well on most of the examples coming from the same distribution. Our lower bounds are much stronger: they hold in the more powerful setting where the algorithm can additionally query a polynomial number of points of its choice and it is required to approximate the target function well on only on a $1/2 + 1/\mathrm{poly}(n)$ fraction of the points coming from the underlying distribution.

Our techniques are also are quite different from those in [15]. First, our learning algorithm is much simpler than theirs — whereas their algorithm uses several submodular optimization algorithms, we only require the aforementioned structural result in order to reduce the problem to a traditional PAC-learning problem. Moreover, our algorithm is more noise tolerant. (We elaborate on this in Section 3.1.) Second, our lower bound is much more technical than theirs — they use a submodular function which is almost "flat" except for a single, large "valley", whereas we require a function which is submodular yet has a *super-polynomial number* of large valleys.

Even in the case of Boolean functions, lower bounds for distributional learning are generally much harder to show than lower bounds for exact learning. For instance, even the extremely simple classes of non-monotone conjunctions or functions having a single positive example are hard for exact learning, and yet they are trivial to PAC-learn. Proving a lower bound for PAC-learning requires exhibiting some degree of fundamental complexity in the class of functions under consideration, especially when one does not restrict the form of the hypothesis function. This phenomenon carries over to the PMAC model as well.

Matroids, Submodular Functions, and Optimization. Optimization problems involving submodular functions have been widely studied in the literature. For example, the minimum of a submodular function can be computed exactly in $\mathrm{poly}(n)$ oracle calls, either by the ellipsoid method [16] or through combinatorial algorithms [31, 18, 20]. Maximizing a (non-monotone) submodular function is NP-hard but, in many settings, constant-factor approximation algorithms have been developed. For example, a $\frac{2}{5}$-approximation has been developed for maximizing any non-negative submodular function [11], and a $(1-1/e)$-approximation algorithm has been derived for maximizing a monotone submodular function subject to a cardinality constraint [13], or an arbitrary matroid constraint [37]. Approximation algorithms for submodular analogues of several other well-known optimization problems have been studied, e.g., submodular set cover [38], submodular sparsest cut [33], submodular vertex cover [19, 14].

2 A Formal Framework

In this section we give formal definitions for the objects and problems studied in this paper. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For $S \subseteq [n]$ and $x \in [n] \setminus S$, let $S + x$ denote $S \cup \{x\}$. Given $S \subseteq [n]$ we will denote by $\chi(S) \in \mathbb{R}^n$ its indicator vector, i.e., $\chi(S)_i$ is 1 if i is in S and 0 otherwise.

2.1 Background and Notation: Submodular Functions and Matroids

We briefly state some standard facts about submodular functions. For a detailed discussion, we refer the reader to Lovász's survey [29], Fujishige's monograph [12] and Schrijver [32].

Definition 1 *A function $f : 2^{[n]} \to \mathbb{R}$ is called* submodular *if*

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T) \quad \textit{for all } S, T \subseteq [n]. \tag{2.1}$$

The function f is called monotone *(or* non-decreasing*) if $f(S) \leq f(T)$ whenever $S \subseteq T \subseteq [n]$.*

An equivalent definition of submodularity is the property of decreasing marginal values. A function $f : 2^{[n]} \to \mathbb{R}$ is submodular iff

$$f(S + x) - f(S) \geq f(T + x) - f(T) \quad \text{for all } S \subseteq T \subseteq [n],\ x \in [n] \setminus T. \tag{2.2}$$

An even more general class of functions is that of subadditive functions.

Definition 2 *A function $f : 2^{[n]} \to \mathbb{R}_+$ is called* subadditive *if*

$$f(S) + f(T) \geq f(S \cup T) \quad \textit{for all } S, T \subseteq [n]. \tag{2.3}$$

A function $f : 2^{[n]} \to \mathbb{R}$ is called L-Lipschitz if $|f(S + x) - f(S)| \leq L$ for all $S \subseteq [n]$ and $x \notin S$.

One manner in which submodular functions arise is as the rank functions of matroids. Further information about matroids can be found in standard references, e.g., Schrijver [32].

Definition 3 *A pair* $\mathbf{M} := (\mathrm{V}, \mathcal{I})$ *is a matroid if* V *is a finite set of elements and* $\mathcal{I} \subseteq 2^{\mathrm{V}}$ *is a non-empty family such that*

- *if* $I \in \mathcal{I}$ *and* $J \subseteq I$*, then* $J \in \mathcal{I}$*, and*
- *if* $I, J \in \mathcal{I}$ *and* $|J| < |I|$*, then there exists an* $i \in I \setminus J$ *such that* $J + i \in \mathcal{I}$*.*

Let $\mathbf{M} = (\mathrm{V}, \mathcal{I})$ be a matroid. The subsets of V in $\mathcal{I}$ are called *independent* and those not in $\mathcal{I}$ are called *dependent*. A maximal independent set is called a base of $\mathbf{M}$. All bases have the same size, which is called the rank of the matroid and denoted $\mathrm{rk}(\mathbf{M})$. Let the function $\mathrm{rank}_{\mathbf{M}} : 2^{\mathrm{V}} \to \mathbb{N}_+$ be defined by

$$\mathrm{rank}_{\mathbf{M}}(S) := \max\{\, |I| \,:\, I \subseteq S,\, I \in \mathcal{I} \,\}.$$

It is well-known that the function rank_M is non-negative, monotone, submodular, and 1-Lipschitz. The converse is also true: any non-negative, integer-valued, monotone, submodular, 1-Lipschitz function with $f(\emptyset) = 0$ is a rank function of some matroid [32].

A well studied family of matroids are partition matroids which are defined as follows. Assume that the sets $A_1, \ldots, A_k \subseteq \mathrm{V}$ are all disjoint and let $u_1, \ldots u_k \in \mathbb{N}$. Let $\mathcal{I} = \{\, I \,:\, |I \cap A_j| \leq u_j \;\forall j \in [k] \,\}$. Then $M = (\mathrm{V}, \mathcal{I})$ is a partition matroid. In this paper we introduce a new general family of matroids which generalizes the family of partition matroids — see Section 4.2

2.2 Our Learning Framework

In this paper we focus on learning in the passive, supervised learning setting. In this setting, we have an instance space $X = \{0,1\}^n$, our data comes from a fixed unknown distribution D over X, and the data is "labeled" by some unknown target function $f^* : X \to \mathbb{R}_+$. A learning algorithm is given a set $\mathcal{S}$ of labeled training examples drawn i.i.d. from D and labeled by f^*. The algorithm performs some computation over the labeled examples $\mathcal{S}$ and outputs a hypothesis $f : X \to \mathbb{R}_+$ that, with high probability, is a good approximation of the target for most points in D. Formally, we define the PMAC learning model as follows.

Definition 4 *Let* $\mathcal{F}$ *be a family of non-negative, real-valued functions with domain* $\{0,1\}^n$ *and let* $g : \mathbb{N} \to \mathbb{R}_+$ *be a function. We say that an algorithm* $\mathcal{A}$ PMAC-learns $\mathcal{F}$ *with an approximation factor* g *if, for any distribution* D *over* $\{0,1\}^n$*, for any target function* $f^* \in \mathcal{F}$*, and for* ϵ *and* δ *sufficiently small we have:*

- *The input to* $\mathcal{A}$ *is a sequence of pairs* $(x_1, y_1), \ldots, (x_\ell, y_\ell)$*, where each* x_i *is chosen independently from distribution* D *and each* $y_i = f^*(x_i)$*.*
- *The input sequence of* $\mathcal{A}$ *has length* $\ell = \mathrm{poly}(n, 1/\epsilon, 1/\delta)$ *and* $\mathcal{A}$ *has running time* $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$*.*
- *The output of* $\mathcal{A}$ *is a function* $f : \{0,1\}^n \to \mathbb{R}$ *that satisfies*

$$\Pr_{(x_1,y_1),\ldots(x_\ell,y_\ell)}\left[\; \Pr_x\left[\, f(x) \leq f^*(x) \leq g(n) \cdot f(x) \,\right] \geq 1 - \epsilon \;\right] \;\geq\; 1 - \delta.$$

The term PMAC stands for "Probably Mostly Approximately Correct". We remark that PMAC-learning with approximation factor 1 is equivalent to PAC-learning. An algorithm which allows the parameter ϵ to take some value $\frac{1}{2} - \Omega(1/\mathrm{poly}(n))$ is said to be a *weak PMAC-learning* algorithm. An algorithm which allows arbitrary parameters $\epsilon > 0$ and $\delta > 0$ is said to be a *strong PMAC-learning* algorithm.

In this paper, we study the following problem. Let $\mathcal{F}$ be the family of all non-negative, monotone, submodular functions. For what ϵ, δ and $g : \mathbb{N} \to \mathbb{R}$ do there exist an algorithm that efficiently PMAC-learns $\mathcal{F}$? How does the answer change if we impose some restrictions on the distribution D or consider a subfamily of $\mathcal{F}$?

3 An $O(\sqrt{n})$-approximation algorithm

In this section we present our upper bounds for efficiently PMAC-learning two very broad families of functions. We give a PMAC-learning algorithm with approximation factor $O(n)$ for learning the family of non-negative, monotone, subadditive functions. We also give a PMAC-learning algorithm with approximation factor $O(\sqrt{n})$ for learning the class of non-negative, monotone, submodular functions.

We start with two lemmas concerning these classes of functions.

Lemma 1 *Let $f : 2^{[n]} \to \mathbb{R}_+$ be a non-negative, monotone, subadditive function. Then there exists a linear function $\hat{f}$ such that $\hat{f}(S) \leq f(S) \leq n\hat{f}(S)$ for all $S \subseteq [n]$.*

Proof: Let $\hat{f}(S) = \frac{\sum_{i \in S} f(\{i\})}{n}$; clearly this is linear. By subadditivity, we have $\hat{f}(S) \geq \frac{f(S)}{n}$, so $f(S) \leq n\hat{f}(S)$. By monotonicity we have $f(S) \geq \max_i f(\{i\})$, so $nf(S) \geq \sum_i f(\{i\})$. This implies $\hat{f}(S) \leq f(S)$, as desired. ■

A stronger result for the class of submodular functions was proven by Goemans et al. [15], using properties of submodular polyhedra and John's theorem on approximating centrally-symmetric convex bodies by ellipsoids [22].

Lemma 2 (Goemans et al. [15]) *Let $f : 2^{[n]} \to \mathbb{R}_+$ be a non-negative, monotone, submodular function with $f(\emptyset) = 0$. Then there exists a function $\hat{f}$ of the form $\hat{f}(S) = \sqrt{w^T \chi(S)}$ where $w \in \mathbb{R}^n_+$ such that $\hat{f}(S) \leq f(S) \leq \sqrt{n}\hat{f}(S)$ for all $S \subseteq [n]$.*

We now use the preceding lemmas in proving our main algorithmic results.

Theorem 3 *Let $\mathcal{F}$ be the class of non-negative, monotone, subadditive functions over $X = 2^{[n]}$. There is an algorithm that PMAC-learns $\mathcal{F}$ with approximation factor $n + 1$. That is, for any distribution D over X, for any ϵ, δ sufficiently small, with probability $1 - \delta$, the algorithm produces a function f that approximates f^* within a multiplicative factor of $n + 1$ on a set of measure $1 - \epsilon$ with respect to D. The algorithm uses $\ell = \frac{16n}{\epsilon} \log\left(\frac{n}{\delta\epsilon}\right)$ training examples and runs in time $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$.*

Theorem 4 *Let $\mathcal{F}$ be the class of non-negative, monotone, submodular functions over $X = 2^{[n]}$. There is an algorithm that PMAC-learns $\mathcal{F}$ with approximation factor $\sqrt{n+1}$. That is, for any distribution D over X, for any ϵ, δ sufficiently small, with probability $1 - \delta$, the algorithm produces a function f that approximates f^* within a multiplicative factor of $\sqrt{n+1}$ on a set of measure $1 - \epsilon$ with respect to D. The algorithm uses $\ell = \frac{16n}{\epsilon} \log\left(\frac{n}{\delta\epsilon}\right)$ training examples and runs in time $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$.*

We remark that our algorithm which proves Theorem 4 is significantly simpler than the algorithm of Goemans et al. [15] which achieves a slightly worse approximation factor in the exact learning model with value queries.

Proof (of Theorem 3). We will show that Algorithm 1 will produce the desired result. For technical reasons, because of the multiplicative error allowed by the PMAC-learning model, we will analyze separately the subset of the instance space where f^* is zero and the subset of the instance space where f^* is non-zero. For convenience, let us define:

$$\mathcal{P} = \{\, S : f^*(S) \neq 0 \,\} \qquad \text{and} \qquad \mathcal{Z} = \{\, S : f^*(S) = 0 \,\}.$$

Algorithm 1 Algorithm for learning subadditive functions.

Input: A sequence of labeled training examples $\mathcal{S} = \{(S_1, f^*(S_1)), (S_2, f^*(S_2)), \ldots (S_\ell, f^*(S_\ell))\}$, where f^* is a subadditive function.

- Let $\mathcal{S}_{\neq 0} = \{(A_1, f^*(A_1)), \ldots, (A_a, f^*(A_a))\}$ be the subsequence of $\mathcal{S}$ with $f^*(A_i) \neq 0 \; \forall i$. Let $\mathcal{S}_0 = \mathcal{S} \setminus \mathcal{S}_{\neq 0}$. Let $\mathcal{U}_0$ be the set of indices defined as

$$\mathcal{U}_0 \;=\; \bigcup_{\substack{i \leq \ell \\ f^*(S_i)=0}} S_i.$$

- For each $1 \leq i \leq a$, let y_i be the outcome of flipping a fair $\{+1, -1\}$-valued coin, each coin flip independent of the others. Let $x_i \in \mathbb{R}^{n+1}$ be the point defined by

$$x_i \;=\; \begin{cases} (\chi(A_i), f^*(A_i)) & (\text{if } y_i = +1) \\ (\chi(A_i), (n+1) \cdot f^*(A_i)) & (\text{if } y_i = -1). \end{cases}$$

- Find a linear separator $u = (w, -z) \in \mathbb{R}^{n+1}$, where $w \in \mathbb{R}^n$ and $z \in \mathbb{R}$, such that u is consistent with the labeled examples $(x_i, y_i) \; \forall i \in [a]$, and with the additional constraint that $w_j = 0 \; \forall j \in \mathcal{U}_0$.

Output: The function f defined as $f(S) = \frac{1}{(n+1)z} w^\mathsf{T} \chi(S)$.

The main idea of our algorithm is to reduce our learning problem to the standard problem of learning a binary classifier (in fact, a linear separator) from i.i.d. samples in the passive, supervised learning setting [24, 36] with a slight twist in order to handle the points in $\mathcal{Z}$. The problem of learning a linear separator in the passive supervised learning setting is one where the instance space is $\mathbb{R}^m$, the samples come from some fixed and unknown distribution D' on $\mathbb{R}^m$, and there is a fixed but unknown target function $c^* : \mathbb{R}^m \to \{-1, +1\}$, $c^*(x) = \text{sgn}(u^\mathsf{T} x)$. The examples induced by D' and c^* are called *linearly separable* since there exists a vector u such that $c^*(x) = \text{sgn}(u^\mathsf{T} x)$.

The linear separator learning problem we reduce to is defined as follows. The instance space is $\mathbb{R}^m$ where $m = n + 1$ and the distribution D' is defined by the following procedure for generating a sample from it. Repeatedly draw a sample $S \subseteq [n]$ from the distribution D until $f^*(S) \neq 0$. Next, flip a fair coin. The sample from D' is

$$\begin{aligned} (\chi(S), f^*(S)) \qquad & (\text{if the coin is heads}) \\ (\chi(S), (n+1) \cdot f^*(S)) \qquad & (\text{if the coin is tails}). \end{aligned}$$

The function c^* defining the labels is as follows: samples for which the coin was heads are labeled $+1$, and the others are labeled -1.

We claim that the distribution over labeled examples induced by D' and c^* is linearly separable in $\mathbb{R}^{n+1}$. To prove this we use Lemma 1 which says that there exists a linear function $\hat{f}(S) = w^\mathsf{T} \chi(S)$ such that $\hat{f}(S) \leq f^*(S) \leq n \cdot \hat{f}(S)$ for all $S \subseteq [n]$. Let $u = ((n + 1/2) \cdot w, -1) \in \mathbb{R}^m$. For any point x in the support of D' we have

$$\begin{aligned} x = (\chi(S), f^*(S)) \qquad &\Longrightarrow \qquad u^\mathsf{T} x = (n + 1/2) \cdot \hat{f}(S) - f^*(S) > 0 \\ x = (\chi(S), (n+1) \cdot f^*(S)) \qquad &\Longrightarrow \qquad u^\mathsf{T} x = (n + 1/2) \cdot \hat{f}(S) - (n+1) \cdot f^*(S) < 0. \end{aligned}$$

This proves the claim. Moreover, this linear function also satisfies $\hat{f}(S) = 0$ for every $S \in \mathcal{Z}$. In particular, $\hat{f}(S) = 0$ for all $S \in \mathcal{S}_0$ and moreover,

$$\hat{f}(\{j\}) \;=\; w_j \;=\; 0 \qquad \text{for every } j \;\in\; \mathcal{U}_D \qquad \text{where} \quad \mathcal{U}_D \;=\; \bigcup_{S_i \in \mathcal{Z}} S_i.$$

Our algorithm is now as follows. It first partitions the training set $\mathcal{S} = \{(S_1, f^*(S_1)), \ldots, (S_\ell, f^*(S_\ell))\}$ into two sets $\mathcal{S}_0$ and $\mathcal{S}_{\neq 0}$, where $\mathcal{S}_0$ is the subsequence of $\mathcal{S}$ with $f^*(S_i) = 0$, and $\mathcal{S}_{\neq 0} = \mathcal{S} \setminus \mathcal{S}_0$. For convenience, let us denote the sequence $\mathcal{S}_{\neq 0}$ as

$$\mathcal{S}_{\neq 0} \;=\; \big((A_1, f^*(A_1)), \ldots, (A_a, f^*(A_a))\big).$$

Note that a is a random variable and we can think of the sets the A_i as drawn independently from D, conditioned on belonging to $\mathcal{P}$. Let

$$\mathcal{U}_0 \;=\; \bigcup_{\substack{i \leq \ell \\ f^*(S_i)=0}} S_i \qquad \text{and} \qquad \mathcal{L}_0 \;=\; \{\, S \,:\, S \subseteq \mathcal{U}_0 \,\}.$$

Using $\mathcal{S}_{\neq 0}$, the algorithm then constructs a sequence $\mathcal{S}'_{\neq 0} \;=\; \big((x_1, y_1), \ldots, (x_a, y_a)\big)$ of training examples for the binary classification problem. For each $1 \leq i \leq a$, let y_i be -1 or 1, each with probability $1/2$. If $y_i = +1$ set $x_i = (\chi(A_i), f^*(A_i))$; otherwise set $x_i = (\chi(A_i), (n+1) \cdot f^*(A_i))$. The last step of our algorithm is to solve a linear program in order to find a linear separator $u = (w, -z)$ where $w \in \mathbb{R}^n$, $z \in \mathbb{R}$ consistent with the labeled examples (x_i, y_i), $i = 1 \leq i \leq a$, with the additional constraints that $w_j = 0$ for $j \in \mathcal{U}_0$. The output hypothesis is $f(S) = \frac{1}{(n+1)z} w^{\mathsf{T}} \chi(S)$. See Algorithm 1.

To prove correctness, note first that the linear program is feasible; this follows from our earlier discussion using the facts (1) $\mathcal{S}'_{\neq 0}$ is a set of labeled examples drawn from D' and labeled by c^* and (2) $\mathcal{U}_0 \subseteq \mathcal{U}_D$. It remains to show that f approximates the target on most of the points. Let $\mathcal{Y}$ denote the set of points $S \in \mathcal{P}$ such that both of the points $(\chi(S), f^*(S))$ and $(\chi(S), (n+1) \cdot f^*(S))$ are correctly labeled by $\operatorname{sgn}(u^{\mathsf{T}} x)$, the linear separator found by our algorithm. It is easy to show that the function $f(S) = \frac{1}{(n+1)z} w^{\mathsf{T}} \chi(S)$ approximates f^* to within a factor $n+1$ on all the points in the set $\mathcal{Y}$. To see this notice that for any point $S \in \mathcal{Y}$, we have

$$\begin{aligned} & w^{\mathsf{T}} \chi(S) - z f^*(S) > 0 \qquad \text{and} \qquad w^{\mathsf{T}} \chi(S) - z(n+1) f^*(S) < 0 \\ \implies \quad & \frac{1}{(n+1)z} w^{\mathsf{T}} \chi(S) \;<\; f^*(S) \;<\; (n+1) \frac{1}{(n+1)z} w^{\mathsf{T}} \chi(S). \end{aligned}$$

So, for any point in $S \in \mathcal{Y}$, the function $f(S) = \frac{1}{(n+1)z} w^{\mathsf{T}} \chi(S)$ approximates f^* to within a factor $n+1$.

Moreover, by design the function f correctly labels as 0 all the examples in $\mathcal{L}_0$. To finish the proof, we now prove two important claims: for our choice of $\ell = \frac{16n}{\epsilon} \log\left(\frac{n}{\delta\epsilon}\right)$, with high probability both $\mathcal{P} \setminus \mathcal{Y}$ and $\mathcal{Z} \setminus \mathcal{L}_0$ have small measure.

Claim 5 *Let $\ell = \frac{16n}{\epsilon} \log\left(\frac{n}{\delta\epsilon}\right)$. Then with probability at least $1 - 2\delta$, the set $\mathcal{P} \setminus \mathcal{Y}$ has measure at most 2ϵ under D.*

Proof. Let $q = 1 - p = \Pr_{S \sim D}[\, S \in \mathcal{P} \,]$. If $q < \epsilon$ then the claim is immediate, since $\mathcal{P}$ has measure at most ϵ. So assume that $q \geq \epsilon$. Let $\mu = \mathbf{E}[\, a \,] = q\ell$. By assumption $\mu > 16n \log(n/\delta\epsilon) \frac{q}{\epsilon}$. Then Chernoff bounds give that

$$\Pr\left[\, a < 8n \log(n/\delta\epsilon) \frac{q}{\epsilon} \,\right] < \exp(-n \log(n/\delta) q/\epsilon) < \delta.$$

So with probability at least $1-\delta$, we have $a \geq 8n\log(qn/\delta\epsilon)\frac{q}{\epsilon}$. By a standard sample complexity argument [36] (which we reproduce in Theorem 24 in Appendix A), with probability at least $1-\delta$, any linear separator consistent with $\mathcal{S}'$ will be inconsistent with the labels on a set of measure at most ϵ/q under D'. In particular, this property holds for the linear separator c computed by the linear program. So for any set S, the conditional probability that either $(\chi(S), f^*(S))$ or $(\chi(S), (n+1)\cdot f^*(S))$ is incorrectly labeled, given that $S \in \mathcal{P}$, is at most $2\epsilon/q$. Thus

$$\Pr[\, S \in \mathcal{P} \;\wedge\; S \notin \mathcal{Y}\,] \;=\; \Pr[\, S \in \mathcal{P}\,] \cdot \Pr[\, S \notin \mathcal{Y} \;\mid\; S \in \mathcal{P}\,] \;\leq\; q \cdot (2\epsilon/q),$$

as required. □

Claim 6 *If $\ell = \frac{16n}{\epsilon}\log\left(\frac{n}{\delta\epsilon}\right)$, then with probability at least $1-\delta$, the set $\mathcal{Z} \setminus \mathcal{L}_0$ has measure at most ϵ.*

Proof. For $k \leq \ell$, define

$$\mathcal{U}_k \;=\; \bigcup_{\substack{i \leq k \\ f^*(S_i)=0}} S_i \qquad \text{and} \qquad \mathcal{L}_k \;=\; \{\, S \,:\, S \subseteq \mathcal{U}_k \,\}\,.$$

So $\mathcal{L}_\ell = \mathcal{L}_0$. By subadditivity, monotonicity, and non-negativity we have $\mathcal{L}_k \subseteq \mathcal{Z}$ for any k. Suppose that, for some k, the set $\mathcal{Z} \setminus \mathcal{L}_k$ has measure at least ϵ. Define $k' = k + \log(n/\delta)/\epsilon$. Then amongst the subsequent examples $S_{k+1}, \ldots, S_{k'}$, the probability that none of them lie in $\mathcal{Z} \setminus \mathcal{L}_k$ is at most $(1-\epsilon)^{\log(n/\delta)/\epsilon} \leq \delta/n$. On the other hand, if one of them does lie in $\mathcal{Z} \setminus \mathcal{L}_k$, then $|\mathcal{U}_{k'}| > |\mathcal{U}_k|$. But $|\mathcal{U}_k| \leq n$ for all k, so this can happen at most n times. Since $\ell \geq n\log(n/\delta)/\epsilon$, with probability at least δ the set $\mathcal{Z} \setminus \mathcal{L}_\ell$ has measure at most ϵ. □

In summary, our algorithm produces a hypothesis f that approximates f^* to within a factor $n+1$ on the set $\mathcal{Y} \cup \mathcal{L}_\ell$. The complement of this set is $(\mathcal{Z} \setminus \mathcal{L}_\ell) \cup (\mathcal{P} \setminus \mathcal{Y})$, which has measure at most 3ϵ, with probability at least $1-3\delta$. ■

The preceding proof was for the class of subadditive functions. The proof for submodular functions is identical, replacing Lemma 1 with Lemma 2.

Proof (of Theorem 4). To learn the class of non-negative, monotone, submodular functions we apply Algorithm 1 with the following changes: (i) in the second step if $y_i = +1$ we set $x_i = (\chi(A_i), f^*(A_i)^2)$ and if $y_i = -1$ we set $x_i = (\chi(A_i), (n+1)\cdot f^*(A_i)^2)$; (ii) we output the function $f(S) = \sqrt{\frac{1}{(n+1)z} w^{\mathsf{T}}\chi(S)}$. To argue correctness we use Lemma 2, which shows that, for any $f \in \mathcal{F}$, the function f^2 can be approximated to within a factor of n by a linear function. The proof of Theorem 3 can then be applied to the family $\{\, f^2 \,:\, f \in \mathcal{F} \,\}$. ■

Potential Improved Guarantees. It is clear from the proofs of Theorem 3 and Theorem 4 that any improvements in the approximation factor for the structural result of Lemma 1 (or Lemma 2) for specific subclasses of subadditive (or submodular) functions immediately translate into PMAC-learning algorithms with improved approximation factors.

3.1 Extensions

The algorithm described for learning subaddive and submodular functions in the PMAC model is quite robust and it can be extended to handle more general cases as well as various forms of noise.

First, we can extend the results in Theorem 3 and Theorem 4 to the more general case where do not even assume that the target function is subadditive (or submodular), but that it is within a factor α of a subadditive (or submodular) function on every point in the instance space. Under this relaxed assumption we are able to achieve the approximation factor $\alpha \cdot (n+1)$ (or $\sqrt{\alpha \cdot (n+1)}$). Specifically:

Theorem 7 *Let $\mathcal{F}$ be the class of non-negative, monotone, subadditive functions over $X = 2^{[n]}$ and let*

$$\mathcal{F}' = \{f : \exists g \in \mathcal{F}, g(S) \leq f(S) \leq \alpha \cdot g(S) \text{ for all } S \subseteq [n]\},$$

for some known $\alpha > 1$. There is an algorithm that PMAC-learns $\mathcal{F}'$ with approximation factor $\alpha(n+1)$. The algorithm uses $\ell = \frac{16n}{\epsilon} \log\left(\frac{n}{\delta\epsilon}\right)$ training examples and runs in time $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$.

Proof: By assumption, there exists $g \in \mathcal{F}$ such that $g(S) \leq f^*(S) \leq \alpha \cdot g(S)$. Combining this with Lemma 1, we get that there exists $\hat{f}(S) = w^{\mathsf{T}}\chi(S)$ such that

$$w^{\mathsf{T}}\chi(S) \leq f^*(S) \leq n \cdot \alpha \cdot w^{\mathsf{T}}\chi(S) \quad \text{for all } S \subseteq [n].$$

In order to learn the class of non-negative monotone submodular functions we apply Algorithm 1 with the following modifications: (1) in the second step if $y_i = +1$ we set $x_i = (\chi(S), f^*(S))$ and if $y_i = -1$ we set $x_i = (\chi(S), \alpha(n+1) \cdot f^*(S))$; (2) we output the function $f(S) = \frac{1}{\alpha(n+1)z} w^{\mathsf{T}}\chi(S)$. It is then easy to show that the distribution over labeled examples induced by D' and c^* is linearly separable in $\mathbb{R}^{n+1}$; in particular, $u = (\alpha(n+1/2) \cdot w, -1) \in \mathbb{R}^{n+1}$ defines a good linear separator. The proof then proceeds as in Theorem 3. ■

Theorem 8 *Let $\mathcal{F}$ be the class of non-negative, monotone, submodular functions over $X = 2^{[n]}$ and let*

$$\mathcal{F}' = \{f : \exists g \in \mathcal{F}, g(S) \leq f(S) \leq \alpha \cdot g(S) \text{ for all } S \subseteq [n]\},$$

for some known $\alpha > 1$. There is an algorithm that PMAC-learns $\mathcal{F}'$ with approximation factor $\sqrt{\alpha \cdot (n+1)}$. The algorithm uses $\ell = \frac{16n}{\epsilon} \log\left(\frac{n}{\delta\epsilon}\right)$ training examples and runs in time $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$.

We can also extend the results in Theorem 3 and Theorem 4 to the *agnostic case* where we assume that the there exists a subadditive (or a submodular) function that agrees with the target on all but an η fraction of the points; note that on the η fraction of the points the target can be arbitrarily far from a subadditive (or a submodular) function. In this case we can still PMAC-learn with a polynomial number of samples $O(\frac{n}{\epsilon^2} \log\left(\frac{n}{\delta\epsilon}\right))$, but using a potentially computationally inefficient procedure.

Theorem 9 *Let $\mathcal{F}$ be the class of non-negative, monotone, subadditive functions over $X = 2^{[n]}$. Let*

$$\mathcal{F}' = \{ f : \exists g \in \mathcal{F} \text{ s.t. } f(S) = g(S) \text{ on more than } 1 - \eta \text{ fraction of the points} \}.$$

There is an algorithm that PMAC-learns $\mathcal{F}'$ with approximation factor $(n+1)$. That is, for any distribution D over X, for any ϵ, δ sufficiently small, with probability $1 - \delta$, the algorithm produces a function f that approximates f^* within a multiplicative factor of $n+1$ on a set of measure $1 - \epsilon - \eta$ with respect to D. The algorithm uses $O(\frac{n}{\epsilon^2} \log\left(\frac{n}{\delta\epsilon}\right))$ training examples.

Proof Sketch: The proof proceeds as in Theorem 3. The main difference is that in the new feature space the best linear separator has error (fraction of mistakes) η. It is well known that even in the agnostic case the number of samples needed to learn a separator of error at most $\eta + \epsilon$ is $O(\frac{n}{\epsilon^2} \log\left(\frac{n}{\delta\epsilon}\right))$ (see Theorem 25 in Appendix A). However, it is NP-hard to minimize the number of mistakes, even approximately [17], so the resulting procedure uses a polynomial number of samples, but it is computationally inefficient. ■

Theorem 10 *Let $\mathcal{F}$ be the class of non-negative, monotone, submodular functions over $X = 2^{[n]}$. Let*

$$\mathcal{F}' \;=\; \{\; f \;:\; \exists g \in \mathcal{F} \text{ s.t. } f(S) = g(S) \;\; \text{on more than } 1 - \eta \text{ fraction of the points } \}\,.$$

There is an algorithm that PMAC-learns $\mathcal{F}'$ with approximation factor $\sqrt{n+1}$. That is, for any distribution D over X, for any ϵ, δ sufficiently small, with probability $1 - \delta$, the algorithm produces a function f that approximates f^* within a multiplicative factor of $\sqrt{n+1}$ on a set of measure $1 - \epsilon - \eta$ with respect to D. The algorithm uses $O(\frac{n}{\epsilon^2} \log\left(\frac{n}{\delta\epsilon}\right))$ training examples.

4 A general lower bound

This section proves a lower bound on the approximation factor achievable by any algorithm which PMAC-learns the class of non-negative, monotone, submodular functions.

Theorem 11 *No algorithm can weakly PMAC-learn the class of non-negative, monotone, submodular functions with approximation factor $o(n^{1/3}/\log n)$.*

The previous theorem gives an information-theoretic hardness result. We can also derive a complexity-theoretic hardness result.

Theorem 12 *Suppose that one-way functions exist. For any constant $\epsilon > 0$, no algorithm can weakly PMAC-learn the class of non-negative, monotone, submodular functions with approximation factor $O(n^{1/3-\epsilon})$, even if the functions are given via polynomial-time algorithms which compute their value on the support of the distribution.*

The proofs of these theorems rely on the following key theorem.

Theorem 13 *Let V be a ground set with $|V| = n$. Let d be a positive integer with $d = o(n^{1/3})$. There exists a family of sets $\mathcal{A} \subseteq 2^V$ and a family of matroids $\mathcal{M} = \{\; \mathbf{M}_{\mathcal{B}} \;:\; \mathcal{B} \subseteq \mathcal{A} \;\}$ with the following properties.*

- *$|\mathcal{A}| = 2^d$ and $|A| \geq n^{1/3}/2$ for every $A \in \mathcal{A}$.*
- *For every $\mathcal{B} \subseteq \mathcal{A}$ and every $A \in \mathcal{A}$, we have*

$$\begin{aligned} \mathrm{rank}_{\mathbf{M}_{\mathcal{B}}}(A) \;&\leq\; 16d \qquad \textit{(if } A \in \mathcal{B}\textit{)} \\ \mathrm{rank}_{\mathbf{M}_{\mathcal{B}}}(A) \;&=\; |A| \qquad \textit{(if } A \in \mathcal{A} \setminus \mathcal{B}\textit{)} \end{aligned}$$

It is worth contrasting the matroid construction of Theorem 13 with a related construction which appeared previously [15], and which has been used to give lower bounds for several problems [33, 19]. That construction sets $\mathcal{A} = \binom{V}{\sqrt{n}}$, and for every $B \in \mathcal{A}$, there is a matroid $\mathbf{M}_B$ for which $\mathrm{rank}_{\mathbf{M}_B}(A)$ is roughly $\log n$ if $A = B$ and roughly $\sqrt{n}$ if $A \in \mathcal{A}$ and $|A \cap B| \geq \log n$. Intuitively, that construction gives a

rank function which is almost like a uniform matroid except that it has a single, large "valley". Our present construction gives rank functions which are almost like a uniform matroid, except that they can have a *super-polynomial number* of large valleys; furthermore, each valley can be independently chosen to be a valley or not, and the resulting function is still a matroid rank function. The proof of Theorem 13 is significantly more involved than the earlier construction.

Proof (of Theorem 11). Let $d = \omega(\log n)$. Let $\mathcal{A}$ and $\mathcal{M}$ be the families constructed by Theorem 13. Let the underlying distribution D on $\{0,1\}^n$ be the uniform distribution on $\mathcal{A}$. (Note that D is not a product distribution on V.) Choose a matroid $\mathbf{M}_{\mathcal{B}} \in \mathcal{M}$ uniformly at random and let the target function be $f^* = \mathrm{rank}_{\mathbf{M}_{\mathcal{B}}}$. Consider any algorithm which attempts to PMAC-learn f^*; note that the algorithm does not know $\mathcal{B}$. For any $A \in \mathcal{A}$ that is not a training example, the algorithm has *no information* about $f^*(A)$, so it cannot determine its value better than randomly guessing between the two possible values $16d$ and $|A|$. The set of non-training examples has measure $1 - 2^{-d+O(\log n)}$. So the expected measure of the set on which the algorithm correctly determines the rank is at most $1/2 + 2^{-d+O(\log n)}$. On the set for which the algorithm did not correctly determine the rank, its approximation factor can be no better than $n^{1/3}/(16d)$. ■

Proof (of Theorem 12). The argument follows Kearns-Valiant [23]. Let $d = n^{\epsilon}$. There exists a family of pseudorandom Boolean functions $F_d = \left\{ f_y : y \in \{0,1\}^d \right\}$, where each function is of the form $f_y : \{0,1\}^d \to \{0,1\}$. Choose an arbitrary bijection between $\{0,1\}^d$ and $\mathcal{A}$. Then each $f_y \in F_d$ corresponds to some subfamily $\mathcal{B} \subseteq \mathcal{A}$, and hence to a matroid rank function $\mathrm{rank}_{\mathbf{M}_{\mathcal{B}}}$. Suppose there is a PMAC-learning algorithm for this family of functions which achieves approximation ratio better than $n^{1/3}/16d$ on a set of measure $1/2 + 1/\operatorname{poly}(n)$. Then this algorithm must be predicting the function f_y on a set of size $1/2 + 1/\operatorname{poly}(n) = 1/2 + 1/\operatorname{poly}(d)$. This is impossible, since the family F_d is pseudorandom. ■

It is possible to show that our lower bound holds even if the algorithm is told the underlying distribution and it is given the ability to query the function on inputs of its choice and even if the queries are adaptive. In other words this lower bound holds even in the PMAC model augmented with value queries.

The remainder of this section proves Theorem 13, and is organized as follows. Section 4.1 shows that a certain uncrossing argument can be used to construct matroids. Section 4.2 then describes a new general family of matroids which arise from the preceding construction. Section 4.3 chooses specific parameters in this construction which yield the lower bound.

4.1 Matroids with Uncrossable Constraints

Let $\mathcal{C}$ be a family of sets and let $f : \mathcal{C} \to \mathbb{Z}$ be a function. Consider the family

$$\mathcal{I} = \{ I : |I \cap C| \leq f(C) \ \forall C \in \mathcal{C} \}. \tag{4.1}$$

For any $I \in \mathcal{I}$, define $T(I) = \{ C \in \mathcal{C} : |I \cap C| = f(C) \}$ to be the set of tight constraints. Suppose that f has the following uncrossing property:

$$\forall I \in \mathcal{I}, \quad C_1, C_2 \in T(I) \quad \Longrightarrow \quad (C_1 \cup C_2 \in T(I)) \ \vee \ (C_1 \cap C_2 = \emptyset). \tag{4.2}$$

Lemma 14 *$\mathcal{I}$ is the family of independent sets of a matroid, if it is non-empty.*

We remark that this lemma implies an alternative proof of a result of Edmonds, which we will not use in this paper.

Corollary 15 (Edmonds [9], Theorem 15) *Let $\mathcal{C}$ be an intersecting family and $f : \mathcal{C} \to \mathbb{R}$ an intersecting-submodular function. Then $\mathcal{I}$ as defined in Eq.* (4.1) *is the family of independent sets of a matroid, if it is non-empty.*

Proof (of Lemma 14). We will show that $\mathcal{I}$ satisfies the required axioms of an independent set family. If $I \subseteq I' \in \mathcal{I}$ then clearly $I \in \mathcal{I}$ also. So suppose that $I \in \mathcal{I}$, $I' \in \mathcal{I}$ and $|I| < |I'|$. Let $C_1, \ldots, C_m$ be the maximal sets in $T(I)$ and let $C^* = \cup_i C_i$. Then $C_i \cap C_j = \emptyset$ for $i \neq j$, otherwise $C_i \cup C_j \in T(I)$ contradicting maximality. Then

$$|I' \cap C^*| \;=\; \sum_{i=1}^{m} |I' \cap C_i| \;\leq\; \sum_{i=1}^{m} f(C_i) \;=\; \sum_{i=1}^{m} |I \cap C_i| \;=\; |I \cap C^*|.$$

Since $|I'| > |I|$, it follows that $|I' \setminus C^*| > |I \setminus C^*|$, so there exists $x \in I' \setminus \big(C^* \cup I\big)$. Then $I + x \in \mathcal{I}$ because for every $C \ni x$ we have $|I \cap C| \leq f(C) - 1$. ■

4.2 A New Matroid Construction

Let $V = [n]$, let $A_1, \ldots, A_k \subseteq V$ be *arbitrary*, and let integers $u_1, \ldots, u_k$ satisfy $0 \leq u_i < |A_i|$. We would like to find a matroid whose rank function has the property that $\mathrm{rank}(A_i) \leq u_i$ for all i, and $\mathrm{rank}(S)$ "large" if S is "far" from any A_i. As mentioned in Section 2.1, if the sets $A_1, \ldots, A_k$ are pairwise disjoint, then the set family $\mathcal{I} \;=\; \{\, I \,:\, |I \cap A_j| \leq u_j \;\forall j \in [k] \,\}$ is is the family of independent sets of a matroid $M = (\mathrm{V}, \mathcal{I})$ (called a partition matroid) whose rank function has the desired property.

Unfortunately, if the sets A_i are not all disjoint, then the family $\mathcal{I}$ is not the independent sets of a matroid. Consider taking $n = 5$, $A_1 = \{1, 2, 3\}$, $A_2 = \{3, 4, 5\}$ and $u_1 = u_2 = 2$. Then both $\{1, 2, 4, 5\}$ and $\{2, 3, 4\}$ are maximal sets in $\mathcal{I}$ but they do not have the same cardinality, which violates the matroid axioms.

However, we can obtain a matroid if we restrict its rank sufficiently, i.e., truncate it. One can show that

$$\mathcal{I} \;=\; \{\, I \,:\, |I \cap A_1| \leq u_1 \;\wedge\; |I \cap A_2| \leq u_2 \;\wedge\; |I| \leq u_1 + u_2 - |A_1 \cap A_2| \,\}$$

is the family of independent sets of a matroid. In this section we show how to generalize this to any k.

For convenience, define the notation $A(J) = \cup_{j \in J} A_j$ for $J \subseteq [k]$. Define the function $f : 2^{[k]} \to \mathbb{Z}$ by

$$f(J) \;=\; |A(J)| - \sum_{j \in J} (|A_j| - u_j).$$

We will prove the following theorem.

Theorem 16 *Let*

$$\mathcal{I} := \{\, I \,:\, |I \cap A(J)| \leq f(J) \;\; \forall J \subseteq [k] \,\}.$$

Then $\mathcal{I}$ is the family of independent sets of a matroid, if it is non-empty.

The following proof, which simplifies our original argument, is due to Jan Vondrák.

Proof: We will apply Lemma 14 to the family $\mathcal{C} = \{ A(J) : J \subseteq [k] \}$ and the function $f' : \mathcal{C} \to \mathbb{Z}$ defined by $f'(C) = \min\{ f(J) : A(J) = C \}$. Fix $I \in \mathcal{I}$ and suppose that C_1 and C_2 are tight, i.e., $|I \cap C_i| = f'(C_i)$. Let J_i satisfy $C_i = A(J_i)$ and $f'(C_i) = f(J_i)$. Define $h_I : 2^{[k]} \to \mathbb{Z}$ by

$$h_I(J) := f(J) - |I \cap A(J)| = |A(J) \setminus I| - \sum_{j \in J}(|A_j| - u_j).$$

Note that $J \mapsto |A(J) \setminus I|$ is a submodular function of J – see Theorem 27 in Appendix B. Since $J \mapsto \sum_{j \in J}(|A_j| - u_j)$ is a modular function of J, we get that h_I is submodular as well.

Since $I \in \mathcal{I}$, we have $h_I \geq 0$. But

$$h_I(J_i) = f(J_i) - |I \cap A(J_i)| = f'(C_i) - |I \cap C_i| = 0,$$

so J_1 and J_2 are minimizers of h_I. It is well-known that the minimizers of any submodular function are closed under union and intersection (see Lemma 28 in Appendix B). So $J_1 \cup J_2$ and $J_1 \cap J_2$ are also minimizers, implying that $A(J_1 \cup J_2) = A(J_1) \cup A(J_2) = C_1 \cup C_2$ is also tight.

This shows that Eq. (4.2) holds, so the theorem follows from Lemma 14. ■

The family $\mathcal{I}$ is non-empty iff $f(J) \geq 0$ for all J. This fails whenever $k > n$ since $f([k]) \leq n - k < 0$. We now modify the preceding construction by introducing a sort of "truncation" operation which allows us to take $k \gg n$.

Definition 5 *Let μ and τ be non-negative integers. The function f is called (μ, τ)-good if*

$$f(J) \geq \begin{cases} 0 & \forall J \subseteq [k],\ |J| < \tau \\ \mu & \forall J \subseteq [k],\ \tau \leq |J| \leq 2\tau - 2. \end{cases}$$

Define $h : 2^{[k]} \to \mathbb{Z}$ by

$$h(J) = \begin{cases} f(J) & \textit{(if } |J| < \tau\textit{)} \\ \mu & \textit{(otherwise).} \end{cases}$$

We now argue that we can replace the function f by h and still obtain a matroid.

Theorem 17 *Suppose that f is (μ, τ)-good. Then the family*

$$\mathcal{I}_h = \{ I : |I \cap A(J)| \leq h(J)\ \forall J \subseteq [k] \}$$

is the family of independent sets of a matroid.

If we assume that $A([k]) = V$ (or if we apply ordinary matroid truncation to reduce the rank to μ) then the family $\mathcal{I}_h$ can be written

$$\mathcal{I}_h = \Big\{ I : |I| \leq \mu \ \wedge\ |I \cap A(J)| \leq f(J)\ \forall J \subseteq [k],\ |J| < \tau \Big\}.$$

Proof (of Theorem 17). Fix $I \in \mathcal{I}_h$. Let J_1 and J_2 satisfy $|I \cap A(J_i)| = h(J_i)$.

Suppose $\max\{|J_1|, |J_2|\} \geq \tau$. Without loss of generality, $|J_1| \geq |J_2|$. Then

$$h(J_1 \cup J_2) = \mu = h(J_1) = |I \cap A(J_1)| \leq |I \cap A(J_1 \cup J_2)|.$$

Otherwise $\max\{|J_1|, |J_2|\} \leq \tau - 1$, so $|J_1 \cup J_2| \leq 2\tau - 2$. We have $|I \cap A(J_i)| = h(J_i) = f(J_i)$ for both i. So, as argued in Theorem 16, we also have $|I \cap A(J_1 \cup J_2)| = f(J_1 \cup J_2)$. But $f(J_1 \cup J_2) \geq h(J_1 \cup J_2)$ since f is (μ, τ)-good.

In both cases we have $|I \cap A(J_1 \cup J_2)| \geq h(J_1 \cup J_2)$, so the desired result follows from Lemma 14. ■

4.3 Proof of Theorem 13

Definition 6 *Let $\mathcal{A} := \{A_1, \ldots, A_k\} \subseteq 2^V$, and let μ, τ and u be non-negative integers with $\tau u \geq 2\mu$. The family $\mathcal{A}$ is called (μ, τ, u)-nice if*

$$\mu/(2d) \;<\; |A_i| \;<\; \mu \qquad \forall i \in [k] \tag{4.3}$$

$$|A_i \cap A(J)| \;\leq\; |J| \cdot u/2 \qquad \forall i \in [k], J \subseteq [k] \setminus \{i\} \text{ s.t. } |J| < \tau \tag{4.4}$$

$$\sum_{j \in J} |A_j| - |A(J)| \;\leq\; |J| \cdot u/2 \qquad \forall J \subseteq [k] \text{ s.t. } |J| \leq 2\tau - 2 \tag{4.5}$$

Proposition 18 *Suppose that $\mathcal{A}$ is (μ, τ, u)-nice. For every $\mathcal{B} \subseteq [k]$ define $f_{\mathcal{B}} : 2^{\mathcal{B}} \to \mathbb{R}$ by*

$$f_{\mathcal{B}}(J) \;=\; |A(J)| - \sum_{j \in J} (|A_j| - u) \qquad \forall J \subseteq \mathcal{B}. \tag{4.6}$$

Then $f_{\mathcal{B}}$ is (μ, τ)-good.

Proof: Suppose that $|J| \leq 2\tau - 2$. Then Eq. (4.5) shows that

$$f_{\mathcal{B}}(J) \;=\; |A(J)| - \sum_{j \in J} (|A_j| - u) \;\geq\; |J| \cdot u/2.$$

This is at least μ whenever $|J| \geq \tau$ since Definition 6 stipulates that $\tau u \geq 2\mu$. Thus $f_{\mathcal{B}}$ is (μ, τ)-good. ■

Lemma 19 *Let d be a positive integer with $d = o(n^{1/3})$. Let the family $\mathcal{A}$ be constructed as follows. Set $k = 2^d$ and construct each set A_i by picking each element of V independently with probability $p = n^{-2/3}$. Then, with constant probability, $\mathcal{A}$ is (μ, τ, u)-nice with parameters*

$$\mu \;=\; n^{1/3} d, \qquad \tau \;=\; n^{1/3}, \quad \textit{and} \quad u \;=\; 16d.$$

Proof (of Theorem 13). Construct the family $\mathcal{A} = \{A_1, \ldots, A_k\}$ randomly as in Lemma 19, where $k = 2^d$. With constant probability, $\mathcal{A}$ is (μ, τ, u)-nice. This implies that each $|A_i| > n^{1/3}/2$. Furthermore, Proposition 18 implies that, for every $\mathcal{B} \subseteq \mathcal{A}$, the function $f_{\mathcal{B}}$ is (μ, τ)-good. So Theorem 17 implies that

$$\mathcal{I}_{\mathcal{B}} \;=\; \left\{ I \;:\; |I| \leq \mu \;\;\wedge\;\; |I \cap A(J)| \leq f_{\mathcal{B}}(J) \;\; \forall J \subseteq \mathcal{B}, |J| < \tau \right\}$$

is the family of independent sets of a matroid, which we call $\mathbf{M}_{\mathcal{B}}$.

It remains to analyze $\mathrm{rank}_{\mathbf{M}_{\mathcal{B}}}(A_i)$ for each $\mathcal{B} \subseteq \mathcal{A}$ and $A_i \in \mathcal{A}$. First, suppose that $A_i \in \mathcal{B}$. The definition of $\mathcal{I}_{\mathcal{B}}$ includes the constraint $|I \cap A_i| \leq f_{\mathcal{B}}(\{i\})$. Thus $\mathrm{rank}_{\mathbf{M}_{\mathcal{B}}}(A_i) \leq f_{\mathcal{B}}(\{i\}) = u = 16d$, as required. Second, we must show that for all $\mathcal{B} \subseteq \mathcal{A}$ and all $A_i \notin \mathcal{B}$ we have $\mathrm{rank}_{\mathbf{M}_{\mathcal{B}}}(A_i) = |A_i|$. This holds iff $A_i \in \mathcal{I}_{\mathcal{B}}$. To prove this, we first observe that $|A_i| \leq \mu$ since $\mathcal{A}$ is (μ, τ, u)-nice. Next, for any $J \subseteq \mathcal{B}$ with $|J| < \tau$, we have

$$|A_i \cap A(J)| \;\leq\; |J| \cdot u/2 \;\leq\; |A(J)| - \sum_{j \in J} (|A_j| - u) \;=\; f_{\mathcal{B}}(J).$$

Here, the first inequality holds due to Eq. (4.4) and the second holds due to Eq. (4.5). Thus A_i satisfies all the constraints and hence $A_i \in \mathcal{I}_{\mathcal{B}}$ ■

It remains to prove Lemma 19. Several steps of the proof involve simple Chernoff bounds, but some steps require the following concentration inequality.

Lemma 20 *Let $A_1, \ldots, A_\ell$ be random subsets of V where each A_i is chosen by including each element of V independently with probability $p < 1/\ell$. Let $Y = \sum_{j=1}^{\ell} |A_j| - \left| \cup_{j=1}^{\ell} A_j \right|$. Then*

$$\begin{aligned} \mathbf{E}[\,Y\,] &\leq n(p\ell)^2/2 \\ \Pr[\,Y \geq t\,] &\leq (1+(p\ell)^2)^n \cdot 2^{-t}. \end{aligned} \tag{4.7}$$

Proof: Clearly $\mathbf{E}[\,A_i\,] = pn$ and $\mathbf{E}\left[\,\cup_{j=1}^{\ell} A_j\,\right] = n(1-(1-p)^\ell)$. Since $(1-p)^\ell \leq 1 - p\ell + (p\ell)^2/2$, we obtain $\mathbf{E}[\,Y\,] \leq n(p\ell)^2/2$, as desired.

To prove Eq. (4.7), we write Y as sum of i.i.d. random variables $Y = \sum_{i=1}^{n} Z_i$, where Z_i are defined as follows. Let $X_1, \ldots, X_n$ be independent random variables distributed according to the binomial distribution with parameters ℓ and p, Then Z_i is defined

$$Z_i = \begin{cases} X_i - 1 & \text{(if } X_i > 0\text{)} \\ 0 & \text{(otherwise).} \end{cases}$$

We now use the moment generating function method. Fix i. Then

$$\begin{aligned} \mathbf{E}\big[2^{Z_i}\big] &= \Pr[\,X_i = 0\,] + \sum_{j=1}^{\ell} 2^{j-1} \Pr[\,X_i = j\,] \\ &= (1-p)^\ell + \frac{1}{2}\sum_{j=1}^{\ell} 2^j \cdot \binom{\ell}{j} \cdot p^j \cdot (1-p)^{\ell-j} \\ &= (1-p)^\ell + \frac{1}{2}\sum_{j=0}^{\ell} \binom{\ell}{j} \cdot (2p)^j \cdot (1-p)^{\ell-j} - \frac{1}{2}\cdot(1-p)^\ell \\ &= \frac{1}{2}(1-p)^\ell + \frac{1}{2}(1+p)^\ell \\ &= \sum_{j=0}^{\lfloor \ell/2 \rfloor} \binom{\ell}{2j} \cdot p^{2j} \leq \sum_{j=0}^{\lfloor \ell/2 \rfloor} \frac{\ell^{2j}}{2^j} \cdot p^{2j} < \sum_{j \geq 0} \left(\frac{(p\ell)^2}{2}\right)^j \\ &\leq \frac{1}{1-(p\ell)^2/2} \leq 1 + (pl)^2. \end{aligned} \tag{4.8}$$

So the Markov inequality implies

$$\Pr[\,Y \geq t\,] = \Pr\big[\,2^Y \geq 2^t\,\big] \leq \frac{\mathbf{E}\big[2^Y\big]}{2^t} = \frac{\mathbf{E}\big[2^Z\big]^n}{2^t}.$$

Combining this together with Eq. (4.8) we obtain

$$\Pr[\,Y \geq t\,] \leq (1+(pl)^2)^n \cdot 2^{-t},$$

as desired. ∎

Proof (of Lemma 19). We show that with high probability Equations (4.3)-(4.5) all hold.

First, consider Eq. (4.3). Since $\mathbf{E}[\,|A_i|\,] = pn = n^{1/3}$, a simple Chernoff bound shows that

$$\Pr\left[\, n^{1/3}/2 < |A_i| < 2n^{1/3} \,\right] \;\geq\; 1 - \exp(-\Omega(n^{1/3})).$$

Since $|\mathcal{A}| = 2^d = o(\exp(n^{1/3}))$, a union bound shows that Eq. (4.3) is satisfied with high probability.

Next, consider Eq. (4.4). Fix $J \subseteq [k]$ with $|J| < \tau$, and fix $i \notin J$. For convenience let $\ell = |J|$. We wish to show that $|A_i \cap A(J)| < \ell u/2$. This is trivial for $\ell = 0$, so assume $\ell \geq 1$. Note that

$$\mathbf{E}[\,|A_i \cap A(J)|\,] = n \cdot \Pr[\,x \in A_i \;\wedge\; x \in A(J)\,] = np(1-(1-p)^\ell).$$

Since $p\ell \leq 1$ we have $1 - p\ell \leq (1-p)^\ell \leq 1 - p\ell/2$, and therefore

$$np^2\ell/2 \;\leq\; \mathbf{E}[\,|A_i \cap A(J)|\,] \;\leq\; np^2\ell.$$

Note that

$$(\ell u/2)/\,\mathbf{E}[\,|A_i \cap A(J)|\,] \;\geq\; (\ell u/2)/(np^2\ell/2) \;=\; 16dn^{1/3} \;\geq\; 2e,$$

so Chernoff bounds show that

$$\Pr[\,|A_i \cap A(J)| \geq \ell u/2\,] \;\leq\; \exp(-\ell u/4).$$

Thus, union bounding over all i and J, the probability that Eq. (4.4) is violated is at most

$$n\sum_{\ell=1}^{\tau-1}\binom{k}{\ell}\exp(-\ell u/4) \;\leq\; n\sum_{\ell=1}^{\tau-1}\exp\big(\ell(\log k - u/4)\big).$$

Since $\log k = d$ and $u = 16d$, we have

$$\exp\big(\ell(\log k - u/4)\big) \;\leq\; \exp\big(-3\log n\big) \;=\; 1/n^3.$$

Thus Eq. (4.4) holds with high probability.

Next, consider Eq. (4.5). Fix $J \subseteq [k]$ with $|J| \leq 2\tau - 2$. As above, let $\ell = |J|$. The inequality is trivial if $\ell \leq 1$ so assume $\ell \geq 2$. Applying Lemma 20,

$$\Pr\left[\sum_{j\in J}|A_j| - |A(J)| > \ell u/2\right] \;\leq\; (1+(p\ell)^2)^n \cdot 2^{-\ell u/2} \;\leq\; \exp\big(\ell(np^2\ell - u/4)\big).$$

This is at most $\exp(-\ell(u/4-2))$ since $np^2\ell < n^{-1/3}\cdot 2\tau = 2$. (Note: this step forces us to choose $p \leq n^{-2/3}$.) Thus, union bounding over all J, the probability that Eq. (4.5) is violated is at most

$$\sum_{\ell=2}^{2\tau-2}\binom{k}{\ell}\exp(-\ell(u/4-2)) \;<\; \sum_{\ell=2}^{2\tau-2}\exp(\ell(\log k - u/4 + 2)) \;<\; n\exp(-2\log n) \;=\; 1/n.$$

Thus Eq. (4.5) holds with high probability. ■

4.4 Rademacher Complexity

We show here how the previous results can be used to provide a lower bound on the Rademacher complexity of monotone submodular functions, a natural measure of the complexity of a class of functions, which we review in Appendix A. Let $\mathcal{F}$ be the class of monotone submodular functions and let the loss function $L_f(x, y)$ to be 0 if $f(x) \leq y \leq \alpha f(x)$ and 1 otherwise. Let $\mathcal{F}_L$ be the the class of functions induced by the original class and the loss function.

Take α to be $n^{1/3}/(2\log^2 n)$. Let D_X be the distribution on $\{0,1\}^n$ that is uniform on the set $\mathcal{A}$ defined in Theorem 11. Let f^* be the target function and let D be the induced distribution over $\{0,1\}^n \times \mathbb{R}$ by the distribution D_X and the target function f^*.

Theorem 21 *For $m = \text{poly}(n, 1/\epsilon, 1/\delta)$, for any sample S of size m from D, $\hat{R}_S(\mathcal{F}_L) \geq 1/4$. Moreover $R_m(\mathcal{F}_L) \geq 1/4$.*

Proof: Let $S = \{(x_1, f^*(x_1)), ..., (x_l, f^*(x_l))\}$ be our i.i.d. set of labeled examples. It is easy to show that $m = \text{poly}(n, 1/\epsilon, 1/\delta)$, then w.h.p. the points x_i are different. Fix a vector $\sigma \in \{-1, 1\}^m$. If $\sigma(i)$ is 1 consider $h_\sigma(x_i) = n^{1/3} + \log^2 n - f^*(x_i)$; if σ_i is -1 consider $h_\sigma(x_i) = f^*(x_i)$. If we then average over σ the quantity $\frac{1}{m}\sum_{i=1}^m \sigma_i \cdot h_\sigma(x_i)$, since for each x_i there are approximately the same number of $+1$ as -1 σ_i values, we get a large constant $\geq 1/4$, as desired. ■

5 Tighter Upper Bounds

In this section we show that matroid rank functions can be strongly PMAC-learned when the distribution on examples is a product distribution. Formally, we show the following result.

Theorem 22 *Let $\mathcal{F}$ be the class of matroid rank functions with ground set V and let D be a product distribution on V. For any sufficiently small $\epsilon > 0$ and $\delta > 0$, there is an algorithm that PMAC-learns $\mathcal{F}$ with approximation factor $1200\log(1/\epsilon)$ using*

$$\ell + \ell' \;=\; 10n^2\log(1/\delta) + n\log(n/\delta)/\epsilon$$

training examples.

If $\mathbf{E}[R] \geq 450\log(1/\epsilon)$ then the approximation factor improves to 8.

Our main technical lemma is the following, which is proven in Appendix D.

Lemma 23 *Let $\mathbf{M} = (V, \mathcal{I})$ be a matroid and let D be a product distribution on V. Let $R = \text{rank}_{\mathbf{M}}(X)$, where X is a sample from D. If $\mathbf{E}[R] \geq 4000$ then, for $\alpha \in [0,1]$,*

$$\Pr[\,|R - \mathbf{E}[R]| \geq (\alpha + 1/4)\,\mathbf{E}[R]\,] \;\leq\; 4e^{-\alpha^2\,\mathbf{E}[R]/12}. \tag{5.1}$$

If $\mathbf{E}[R] \leq 500\log(1/\epsilon)$ then

$$\Pr[\,R \geq 1200\log(1/\epsilon)\,] \;\leq\; \epsilon. \tag{5.2}$$

A special case of this result is the following fact in linear algebra. Fix an matrix A over any field. Construct a random submatrix by selecting the i^{th} column of A with some probability p_i, where these selections are made independently. Then the rank of the resulting submatrix is highly concentrated around

its expectation. Although this fact is very natural, we are unaware of it being previously used, even though its proof is a prototypical application of Talagrand's inequality. In matrix analysis, Talagrand's inequality has primarily been used for studying the concentration of the largest eigenvalue of matrices with random entries [1].

Theorem 22 is proven by analyzing Algorithm 2, using Lemma 23.

Algorithm 2 Algorithm for learning matroid rank functions under a product distribution. Its input is a sequence of labeled training examples $(S_1, f^*(S_1)), (S_2, f^*(S_2)), \ldots$, where f^* is a matroid rank function.

- Let $\mu = \sum_{i=1}^{\ell} f^*(S_i)/\ell$, where $\ell = 10n^2 \log(1/\delta)$.
- *Case 1:* If $\mu \geq 450 \log(1/\epsilon)$, then return the constant function $f = \mu/4$.
- *Case 2:* If $\mu < 450 \log(1/\epsilon)$, then use $\ell' = n \log(n/\delta)/\epsilon$ additional samples to compute the set

$$\mathcal{U} = \bigcup_{\substack{\ell \leq i \leq \ell+\ell' \\ f^*(S_i)=0}} S_i.$$

  Return the function f where $f(A) = 0$ if $A \subseteq \mathcal{U}$ and $f(A) = 1$ otherwise.

To analyze this algorithm, let us first consider the estimate μ. Note that $0 \leq R \leq n$. Then a Hoeffding bound implies that, with probability at least $1 - \delta$,

$$\begin{aligned} \mu \geq 450 \log(1/\epsilon) \quad &\Longrightarrow \quad \mathbf{E}[R] \geq 400 \log(1/\epsilon) \quad \text{and} \quad \tfrac{5}{6}\mathbf{E}[R] \leq \mu \leq \tfrac{4}{3}\mathbf{E}[R] \\ \mu < 450 \log(1/\epsilon) \quad &\Longrightarrow \quad \mathbf{E}[R] \leq 500 \log(1/\epsilon). \end{aligned}$$

Case 1: Assume that $\mathbf{E}[R] \geq 400 \log(1/\epsilon)$; this holds with probability at least $1 - \delta$. If ϵ is sufficiently small then $\mathbf{E}[R] \geq 4000$, so Eq. (5.1) implies

$$\begin{aligned} \Pr[\mu/4 \leq R \leq 2\mu] &\geq \Pr\left[\tfrac{1}{3}\mathbf{E}[R] \leq R \leq \tfrac{5}{3}\mathbf{E}[R]\right] \\ &\geq 1 - \Pr[|R - \mathbf{E}[R]| \geq (2/3)\mathbf{E}[R]] \\ &\geq 1 - 4e^{-\mathbf{E}[R]/100} \geq 1 - \epsilon. \end{aligned}$$

Therefore, with confidence at least $1 - \delta$, the algorithm achieves approximation factor 8 on all but an ϵ fraction of the distribution.

Case 2: Assume that $\mathbf{E}[R] < 500 \log(1/\epsilon)$; this holds with probability at least $1 - \delta$. The argument mirrors a portion of the proof of Theorem 3. We have

$$\mathcal{P} = \{ S : f^*(S) \neq 0 \} \qquad \text{and} \qquad \mathcal{Z} = \{ S : f^*(S) = 0 \}.$$

Let S be a sample from D. Let $\mathcal{E}$ be the event that S violates the inequality

$$f(S) \leq f^*(S) \leq 1200 \log(1/\epsilon) f(S).$$

Clearly $\Pr[\mathcal{E}] = \Pr[\mathcal{E} \wedge S \in \mathcal{Z}] + \Pr[\mathcal{E} \wedge S \in \mathcal{P}]$. First suppose that $S \in \mathcal{Z}$, i.e., $f^*(S) = 0$. Let $\mathcal{L} = \{ S : S \subseteq \mathcal{U} \}$. Since $f(T) = 0$ for all $T \in \mathcal{L}$, the event $\mathcal{E} \wedge S \in \mathcal{Z}$ holds only if $S \in \mathcal{Z} \setminus \mathcal{L}$. By Claim 6, the set $\mathcal{Z} \setminus \mathcal{L}$ has measure at most ϵ, so $\Pr[\mathcal{E} \wedge S \in \mathcal{Z}] \leq \epsilon$.

Now suppose that $S \in \mathcal{P}$, i.e., $f^*(S) \geq 1$. Since every $T \subseteq \mathcal{U}$ satisfies $f^*(T) = 0$, we have $S \not\subseteq \mathcal{U}$ and hence $f(S) = 1$. Therefore the event $\mathcal{E} \wedge S \in \mathcal{P}$ occurs only when $f^*(S) > 1200 \log(1/\epsilon)$. By Eq. (5.2), this happens with probability at most ϵ. This concludes the proof of Theorem 22.

6 Conclusions and Open Questions

In this paper we study submodular function learning in the traditional distributional learning setting. We prove polynomial upper and lower bounds on the approximability guarantees achievable in the general case by using only a polynomial number of examples. We also provide improved analyses for specific classes of submodular functions.

Our work combines central issues in optimization (submodular functions and matroids) with central issues in learning (learnability of natural but complex classes of functions in a distributional setting). Our analysis brings a twist on the usual learning theory models and uncovers some interesting extremal properties of submodular functions, which are likely to be useful in other contexts as well.

Open Questions It would be interesting to close the gap between the $O(n^{1/2})$ upper bound in Theorem 4 and the $\tilde{\Omega}(n^{1/3})$ lower bound in Theorem 11. We suspect that the lower bound can be improved to $\tilde{\Omega}(n^{1/2})$. If such an improved lower bound is possible, the matroids or submodular functions used in its proof are likely to be very interesting. It would be also be interesting to use the approach in our general $\sqrt{n}$-upper bound for simplifying the analysis in the upper bound of Goemans et al. [15].

The algorithm in Section 5 applies to matroid rank functions. It trivially extends to L-Lipschitz functions for any constant L. What if $L \geq n$?

Are there particular subclasses of submodular functions for which one can PMAC-learn with approximation ratio better than $O(\sqrt{n})$, perhaps under additional distributional assumptions? Can one PMAC-learn other natural classes of real-valued functions, with good approximation ratios?

Acknowledgements

We would like to thank Avrim Blum and Jan Vondrák for insightful and stimulating discussions. We also thank Steve Hanneke, Lap Chi Lau, Alex Samorodnitsky, Mohit Singh, Santosh Vempala, and Van Vu for helpful discussions.

References

[1] N. Alon, M. Krivelevich, and V. H. Vu. On the concentration of eigenvalues of random symmetric matrices. *Israel Journal of Mathematics*, 131:259–267, 2002.

[2] N. Alon and J. Spencer. *The Probabilistic Method*. Wiley, 2000.

[3] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[4] M. F. Balcan, A. Blum, and Y. Mansour. Item pricing for revenue maxmimization. In *ACM Conference on Electronic Commerce*, 2009.

[5] A. Blum, C. Burch, and J. Langford. On learning monotone boolean functions. In *FOCS*, 1998.

[6] D. Dachman-Soled, T. Malkin K. Lee, R. Servedio, A. Wan, and H. Wee. Optimal cryptographic hardness of learning monotone functions. In *ICALP*, 2008.

[7] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

[8] S. Dobzinski, N. Nisan, and M. Schapira. Truthful Randomized Mechanisms for Combinatorial Auctions. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 644–652, 2006.

[9] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In R. Guy, H. Hanani, N. Sauer, and J. Schönheim, editors, *Combinatorial Structures and Their Applications*, pages 69–87. Gordon and Breach, 1970.

[10] O. Ekin, P. L. Hammer, and U. N. Peled. Horn functions and submodular boolean functions. *Theoretical Computer Science*, 175(2):257–270, 1997.

[11] U. Feige, V. Mirrokni, and J. Vondrák. Maximizing non-monotone submodular functions. In *FOCS*, 2007.

[12] S. Fujishige. *Submodular Functions and Optimization*. Elsevier, 2005.

[13] L. A. Wolsey G. L. Nemhauser and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14, 1978.

[14] G. Goel, C. Karande, P. Tripathi, and L. Wang. Approximability of combinatorial problems with multi-agent submodular cost functions. In *FOCS*, 2009.

[15] M. Goemans, N. Harvey, S. Iwata, and V. Mirrokni. Approximating submodular functions everywhere. In *ACM-SIAM Symposium on Discrete Algorithms*, 2009.

[16] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer Verlag, 1993.

[17] V. Guruswami and P. Raghavendra. Hardness of Learning Halfspaces with Noise. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006.

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

[19] S. Iwata and K. Nagano. Submodular function minimization under covering constraints. In *FOCS*, 2009.

[20] S. Iwata and J. Orlin. A simple combinatorial algorithm for submodular function minimization. In *SODA*, 2009.

[21] Svante Janson, Tomasz Łuczak, and Andrzej Ruciński. *Random Graphs*. Wiley-Interscience, 2000.

[22] F. John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays, presented to R. Courant on his 60th Birthday, January 8, 1948*, 1948.

[23] M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

[24] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[25] A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. In *UAI*, 2005.

[26] A. Krause and C. Guestrin. Beyond convexity: Submodularity in machine learning, 2008. http://www.select.cs.cmu.edu/tutorials/icml08submodularity.html.

[27] A. Krause and C. Guestrin. Intelligent information gathering and submodular function optimization, 2009. http://submodularity.org/ijcai09/index.html.

[28] B. Lehmann, D. J. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55:270–296, 2006.

[29] L. Lovász. Submodular functions and convexity. *Mathematical Programmming: The State of the Art*, 1983.

[30] M. Narasimhan and J. Bilmes. Local search for balanced submodular clusterings. In *Twentieth International Joint Conference on Artificial Intelligence*, 2007.

[31] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80:346–355, 2000.

[32] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2004.

[33] Z. Svitkina and L. Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. In *FOCS*, 2008.

[34] M. Talagrand. Concentration of measure and isoperimetric inequalites in product spaces. *Publications Mathématiques de l'I.H.É.S.*, 81(1):73–205, December 1995.

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

[36] V. N. Vapnik. *Statistical Learning Theory*. Wiley and Sons, 1998.

[37] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, 2008.

[38] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2:385–393, 1982.

A Standard Sample Complexity Results

We state here several known sample complexity bounds. We start with the classic VC-dimension based bounds. See, e.g., [7, 3].

Theorem 24 *Let C be a set of functions from X to $\{-1,1\}$ with finite VC-dimension $D \geq 1$. Let D be an arbitrary, but fixed probability distribution over X and let c^* be an arbitrary target function. For any ϵ, $\delta > 0$, if we draw a sample from D of size $N(\epsilon,\delta) = \frac{1}{\epsilon}\left(4D\log\left(\frac{1}{\epsilon}\right) + 2\log\left(\frac{2}{\delta}\right)\right)$, then with probability $1-\delta$, all hypotheses with error $\geq \epsilon$ are inconsistent with the data.*

Theorem 25 *Suppose that C is a set of functions from X to $\{-1,1\}$ with finite VC-dimension $D \geq 1$. For any distribution D over X, any target function (not necessarily in C), and any ϵ, $\delta > 0$, if we draw a sample from D of size*

$$m(\epsilon,\delta,D) = \frac{64}{\epsilon^2}\left(2D\ln\left(\frac{12}{\epsilon}\right) + \ln\left(\frac{4}{\delta}\right)\right),$$

then with probability at least $1-\delta$*, we have* $|err(h) - \widehat{err}(h)| \leq \epsilon$ *for all* $h \in C$*.*

Definition 7 *For a sample* $S = \{z_1, ..., z_m\}$ *generated by a distribution* D *on a set* Z *and a real-valued function class* $\mathcal{F}$ *with a domain* Z*, the* empirical Rademacher complexity *of* $\mathcal{F}$ *is the random variable:*

$$\hat{R}_S(\mathcal{F}) = \mathbf{E}_\sigma \left[\sup_{h \in \mathcal{F}} \frac{1}{m} \sum_{i}^{m} \sigma_i \cdot h(z_i) \right]$$

where $\sigma = (\sigma_1, ..., \sigma_m)$ *are independent* $\{-1, +1\}$*-valued (Rademacher) random variables.*

The (distributional) Rademacher complexity of $\mathcal{F}$ *is:*

$$R_m(\mathcal{F}) = \mathbf{E}_S[R_S(\mathcal{F})].$$

Let $Z = X \times Y$, where X is the instance space and Y is the label space. With these notations we have:

Theorem 26 *Fix* $\mathcal{F}$ *be a class of functions mapping from* Z *to* $[a, a+1]$*. Let* $S = \{z_1, ..., z_m\}$ *be an i.i.d. sample from* D*. Then with probability* $1-\delta$ *over the random draws of sample of size* m *we have:*

$$\mathbf{E}_D[f(z)] \leq \hat{\mathbf{E}}_D[f(z)] + R_l(\mathcal{F}) + \sqrt{\frac{\ln(2/\delta)}{2m}} \leq \hat{\mathbf{E}}_D[f(z)] + \hat{R}_S(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2m}}.$$

B Standard Facts about Submodular Functions

Theorem 27 *Given a finite universe* U*, let* S_1*,* S_2*, . . . ,* S_n *be subsets of* U*. Define* $f : 2^{[n]} \to \mathbb{R}_+$ *by*

$$f(A) = |\cup_{i \in A} S_i| \quad \textit{for} \quad A \subseteq [n].$$

Then f *is monotone and submodular. More generally, for any non-negative weight function* $w : U \to \mathbb{R}_+$*, the function* f *defined by*

$$f(A) = w\left(\cup_{i \in A} S_i\right) \quad \textit{for} \quad A \subseteq [n]$$

is monotone and submodular.

Lemma 28 *The minimizers of any submodular function are closed under union and intersection.*

Proof: Assume that J_1 and J_2 are minimizers for f. By submodularity we have

$$f(J_1) + f(J_2) \geq f(J_1 \cap J_2) + f(J_1 \cup J_2).$$

We also have

$$f(J_1 \cap J_2) + f(J_1 \cup J_2) \geq f(J_1) + f(J_2),$$

so $f(J_1) = f(J_2) = f(J_1 \cap J_2) = f(J_1 \cup J_2)$, as desired. ■

C Special Cases of Theorem 17

The matroid construction of Theorem 17 has several interesting special cases.

Partition Matroids Suppose that the sets $A_1, \ldots, A_k$ are all disjoint. We claim that the matroid $\mathcal{I}$ defined above is a partition matroid. To see this, note that $f(J) = \sum_{j \in J} u_j$, since the A_j's are disjoint, so f is a modular function. Similarly, $|I \cap A(J)|$ is a modular function of J. Thus, whenever $|J| > 1$, the constraint $|I \cap A(J)| \leq f(J)$ is redundant — it is implied by the constraints $|I \cap A_j| \leq u_j$ for $j \in J$. So we have

$$\mathcal{I} \;=\; \{\, I \,:\, |I \cap A(J)| \leq f(J)\; \forall J \subseteq [k] \,\} \;=\; \{\, I \,:\, |I \cap A_j| \leq u_j\; \forall j \in [k] \,\},$$

which is a partition matroid.

Complements of Hamming Balls. We are given sets $A_1, \ldots, A_k$ and values $u_1, \ldots, u_k$. Let $\tau = 2$. Note that for any pair $J = \{i, j\}$, we have $f(J) = u_i + u_j - |A_i \cap A_j|$. So define

$$\mu \;=\; \min_{J \subseteq [k],\, |J|=2} f(J) \;=\; \min_{i,j \in [k]} (u_i + u_j - |A_i \cap A_j|).$$

Then f is $(\mu, 2)$-good. So the family

$$\mathcal{I}_h \;=\; \{\, I \,:\, |I| \leq \mu \;\wedge\; |I \cap A_j| \leq u_j\; \forall j \in [k] \,\}$$

is the family of independent sets of a matroid. We call such a matroid a bumpy matroid.

As a special case, suppose that we are given $A_1, \ldots, A_k$ and integers $d_1, \ldots, d_k$ such that

$$\begin{aligned} |A_j| &= \mu && \forall j \in [k] \\ 0 \leq d_j &\leq |A_j| && \forall j \in [k] \\ |A_i \oplus A_j| &\geq 2(d_i + d_j) && \forall i, j \in [k]. \end{aligned}$$

Define $u_j = \mu - d_j$. Then

$$|A_i \oplus A_j| \geq 2(d_i + d_j) \quad\Longrightarrow\quad |A_i \cap A_j| \leq \mu - d_i - d_j \quad\Longrightarrow\quad \mu \leq u_i + u_j - |A_i \cap A_j|.$$

We obtain that

$$\mathcal{B} \;=\; \{\, B \,:\, |B| = \mu \;\wedge\; |B \oplus A_j| \geq 2d_j\; \forall j \in [k] \,\}$$

is the family of bases of a matroid. So the bases are the sets of weight μ not contained in any Hamming ball of radius $2d_j - 1$ centered at A_j.

Paving Matroids Suppose that we are given $A_1, \ldots, A_k$ such that

$$\begin{aligned} |A_j| &= \mu && \forall j \in [k] \\ |A_i \oplus A_j| &\geq 4 && \forall i, j \in [k]. \end{aligned}$$

As argued above,

$$\mathcal{B} \;=\; \{\, B \,:\, |B| = \mu \;\wedge\; |B \oplus A_j| \geq 2\; \forall j \in [k] \,\}$$

is the family of bases of a matroid. Such a matroid is called a paving matroid.

D Proof of Lemma 23

We require the following result [2, §7.7], which follows from Talagrand's inequality [34].

Let D be a product distribution on V. Let $f : 2^V \to \mathbb{R}$ be a 1-Lipschitz function. Let c be a function $c : \mathbb{N} \to \mathbb{N}$. The function f is called c-certifiable if whenever $f(X) \geq s$ there exists $I \subseteq V$ with $|I| \leq c(s)$ such that all $Y \subseteq V$ with $X \cap I = Y \cap I$ also have $f(Y) \geq s$.

Theorem 29 (Talagrand) *Let X be drawn from distribution D and let $R = f(X)$. For all b and $t \geq 0$,*

$$\Pr\left[\, R \leq b - t\sqrt{c(b)} \,\right] \cdot \Pr\left[\, R \geq b \,\right] \;\leq\; e^{-t^2/4}.$$

Proof (of Lemma 23). Clearly, the rank function $\mathrm{rank}_{\mathbf{M}}$ is 1-Lipschitz. Moreover, whenever $\mathrm{rank}_{\mathbf{M}}(X) \geq s$, there exists an independent set $I \subseteq X$ with $|I| = s$. Hence all sets $Y \supseteq I$ have $\mathrm{rank}_{\mathbf{M}}(Y) \geq s$. This implies that $\mathrm{rank}_{\mathbf{M}}$ is c-certifiable with $c(s) = s$.

Let M be any median of R and let $\lambda \geq 0$. Taking $b = M$ and $t = \lambda/\sqrt{M}$, Theorem 29 implies

$$\Pr\left[\, R \leq M - \lambda \,\right] \leq 2e^{-\lambda^2/4M}.$$

Taking $b = M + \lambda$ and $t = \lambda/\sqrt{M+\lambda}$, Theorem 29 implies

$$\Pr\left[\, R \geq M + \lambda \,\right] \leq 2e^{-\lambda^2/4(M+\lambda)}.$$

This shows that R is tightly concentrated around M. The following claim implies that R is also tightly concentrated around $\mathbf{E}\left[\, R \,\right]$.

Claim 30 $\left|\mathbf{E}\left[\, R \,\right] - M\right| \leq 15\sqrt{\mathbf{E}\left[\, R \,\right]} + 32.$

Proof. This is a standard calculation; see, e.g., [21, §2.5]. Note that

$$\Pr\left[\, |R - M| \geq \lambda \,\right] \;\leq\; \begin{cases} 4e^{-\lambda^2/8M} & 0 \leq \lambda \leq M \\ 4e^{-\lambda/8} & \lambda \geq M. \end{cases}$$

Then

$$\begin{aligned} \left|\mathbf{E}\left[\, R \,\right] - M\right| \;&\leq\; \mathbf{E}\left[\, |R - M| \,\right] \\ &=\; \int_0^\infty \Pr\left[\, |R - M| \geq \lambda \,\right] d\lambda \\ &=\; \int_0^M 4e^{-\lambda^2/8M}\, d\lambda \;+\; \int_M^\infty 4e^{-\lambda/8}\, d\lambda \\ &\leq\; 4\sqrt{2\pi M} + 32e^{-M/8}. \end{aligned}$$

Since $R \geq 0$ we have $0 \leq M \leq 2\,\mathbf{E}\left[\, R \,\right]$ (by Markov's inequality), and the desired result follows. □

Thus,

$$\begin{aligned} \Pr\left[\, |R - \mathbf{E}\left[\, R \,\right]| \geq \lambda \,\right] \;&\leq\; \Pr\left[\, |R - M| \geq \lambda - |M - \mathbf{E}\left[\, R \,\right]| \,\right] \\ &\leq\; \Pr\left[\, |R - M| \geq \lambda - 15\sqrt{\mathbf{E}\left[\, R \,\right]} - 32 \,\right]. \end{aligned}$$

To prove Eq. (5.1), let $\lambda = \alpha \mathbf{E}[R] + 15\sqrt{\mathbf{E}[R]} + 32$. Then

$$\begin{aligned}\Pr[|R - \mathbf{E}[R]| \geq \lambda] &\leq \Pr[R \leq M - \alpha \mathbf{E}[R]] + \Pr[R \geq M + \alpha \mathbf{E}[R]] \\ &\leq 2e^{-(\alpha \mathbf{E}[R])^2/4M} + 2e^{-(\alpha \mathbf{E}[R])^2/4(M+\alpha \mathbf{E}[R])} \\ &\leq 4e^{-\alpha^2 \mathbf{E}[R]/12},\end{aligned}$$

since $M \leq 2\mathbf{E}[R]$. Now $15\sqrt{\mathbf{E}[R]} + 32 \leq \mathbf{E}[R]/4$ since $\mathbf{E}[R] \geq 4000$. Thus

$$\Pr[|R - \mathbf{E}[R]| \geq (\alpha + 1/4)\mathbf{E}[R]] \leq \Pr[|R - \mathbf{E}[R]| \geq \lambda] \leq 4e^{-\alpha^2 \mathbf{E}[R]/12}.$$

To prove Eq. (5.2), let $b = 1200\log(1/\epsilon)$ and $t = 4\sqrt{\log(1/\epsilon)}$. Then $b - t\sqrt{c(b)} \geq 1000\log(1/\epsilon) \geq 2\mathbf{E}[R]$. By Markov's inequality, $\Pr\left[R \leq b - t\sqrt{c(b)}\right] \geq 1/2$. So, by Theorem 29, $\Pr[R \geq b] \leq 2\exp(-t^2/4) \leq \epsilon$. ■

E Boolean Submodular Functions

Proposition 31 *The class of monotone, Boolean-valued, submodular functions is efficiently PMAC-learnable with approximation factor* 1.

Proof: Let $f : 2^{[n]} \to \{0, 1\}$ be an arbitrary monotone, submodular function. We claim that f is either constant or a monotone disjunction.

By (2.2), we have $f(T \cup \{x\}) - f(T) \leq f(S \cup \{x\}) - f(S)$, for $S \subseteq T \subseteq [n]$, $x \in [n] \setminus T$. Considering $S = \emptyset$, we get

$$f(T \cup \{x\}) - f(T) \leq f(\{x\}) \quad \forall T \subseteq [n], x \in [n] \setminus T.$$

This implies that if $f(\{x\}) = 0$, then $f(T \cup \{x\}) \leq f(T)$ for all T and all x and by monotonicity, we get that if $f(\{x\}) = 0$, then $f(T \cup \{x\}) = f(T)$ for all T and all x. Also by monotonicity, if $f(\{x\}) = 1$ then $f(T) = 1$ for all T such that $x \in T$. This then implies that $f(S) = \bigvee_{x_i \in S} x_i$ for all $S \in 2^{[n]}$.

This proves the claim. It is well known that the class of disjunctions is easy to learn in the supervised learning setting [24, 36]. ■

Non-monotone, Boolean, submodular functions need not be disjunctions. For example, consider the function $f(S) = 0$ if $S = \emptyset$ or $S = [n]$ and $f(S) = 1$; it is submodular, but not a disjunction. However, it turns out that any submodular boolean functions is a 2-DNF [10]. It is well known that such functions are efficiently PAC-learnable. We summarize this discussion as follows.

Proposition 32 *The class of Boolean, submodular functions is efficiently PMAC-learnable with approximation factor* 1.