Sketching Valuation Functions

Ashwinkumar Badanidiyuru * Shahar Dobzinski* Hu Fu* Robert Kleinberg*
Noam Nisan† Tim Roughgarden‡

October 4, 2011

Abstract

Motivated by the problem of querying and communicating bidders' valuations in combinatorial auctions, we study how well different classes of set functions can be sketched. More formally, let f be a function mapping subsets of some ground set $[n]$ to the non-negative real numbers. We say that f' is an α-sketch of f if for every set S, the value $f'(S)$ lies between $f(S)/\alpha$ and $f(S)$, and f' can be specified by poly(n) bits.

We show that for every subadditive function f there exists an α-sketch where $\alpha = n^{1/2} \cdot O(\text{polylog}(n))$. Furthermore, we provide an algorithm that finds these sketches with a polynomial number of demand queries. This is essentially the best we can hope for since:

1. We show that there exist subadditive functions (in fact, XOS functions) that do not admit an $o(n^{1/2})$ sketch. (Balcan and Harvey [3] previously showed that there exist functions belonging to the class of substitutes valuations that do not admit an $O(n^{1/3})$ sketch.)

2. We prove that every deterministic algorithm that accesses the function via *value* queries only cannot guarantee a sketching ratio better than $n^{1-\epsilon}$.

We also show that coverage functions, an interesting subclass of submodular functions, admit arbitrarily good sketches.

Finally, we show an interesting connection between sketching and learning. We show that for every class of valuations, if the class admits an α-sketch, then it can be α-approximately learned in the PMAC model of Balcan and Harvey. The bounds we prove are only information-theoretic and do not imply the existence of computationally efficient learning algorithms in general.

1 Introduction

On a finite set N, where $|N| = n$, a set function $f : 2^N \to \mathbb{R}_+$ is said to be *subadditive* if $f(S) + f(T) \geq f(S \cup T)$ for all sets S, T. In this paper we consider functions that are monotone, i.e., $f(T) \geq f(S)$ for all $S \subseteq T$, and normalized, $f(\emptyset) = 0$.

Subadditive functions arise naturally in economics as they capture the notion of complement freeness in a fairly general sense. For example, a buyer facing multiple items has a subadditive valuation function if having two sets of items simultaneously does not generate extra value for him. Combinatorial auctions with this type of valuation functions have been extensively studied; see [5, 9, 12] for example. In [16], a hierarchy of complement free functions was defined, with subadditive functions being the most general class. In particular, it strictly contains *submodular* functions.

Submodular functions correspond to the economic concept of "diminishing returns". Formally, a set function f is submodular if $f(S \cup \{j\}) - f(S) \geq f(T \cup \{j\}) - f(T)$, for all $S \subseteq T$ and $j \in N \setminus T$. Such functions arise extensively in combinatorial optimization [17], economics [21], social networks [18], and recently machine learning [14, 15].

It is often of interest to communicate such set functions among different parties; however, set functions in general need 2^n values to describe — a value for each possible bundle. Since the property of complement freeness entails restrictions among the function values, one may naturally ask if a reasonable estimation of such a function can be obtained with much less information from the function, at the loss of some exactitude. We call such an estimation a *sketch*.

---

*Department of Computer Science, Cornell University. Research supported by NSF awards CCF-0643934, IIS-0905467, and AF-0910940, AFOSR grant FA9550-09-1-0100, a Microsoft Research New Faculty Fellowship, a Google Research Grant, and an Alfred P. Sloan Foundation Fellowship. Email: {ashwin85,hufu,rdk,shahar}@cs.cornell.edu

†School of Computer Science and Engineering, Hebrew University of Jerusalem. Supported by a grant from the Israeli Science Foundation (ISF), and by the Google Inter-university center for Electronic Markets and Auctions. Email: noam@cs.huji.ac.il

‡Department of Computer Science, Stanford University, 462 Gates Building, 353 Serra Mall, Stanford, CA 94305. Supported in part by NSF grant CCF-1016885, an ONR PECASE Award, and an AFOSR MURI grant. Email: tim@cs.stanford.edu.

More formally, we say that $g : 2^N \to \mathbb{R}_+$ is an *α-approximation* of $f : 2^N \to \mathbb{R}_+$ if for every set S we have that, $\frac{f(S)}{\alpha} \le g(S) \le f(S)$. We say that g is an *α-sketch* if in addition it can be represented by $poly(n)$ bits[1]. Of course, we are not interested only in proving existence of sketches that provide a good approximation ratio, but would also like to construct these sketches efficiently.

Goemans et al. [13] showed that, when f is submodular, an $\tilde{O}(\sqrt{n})$-sketch of f can be obtained by querying f's value at polynomially many subsets of S (i.e., using *$poly(n)$ value queries*). Their construction is essentially tight: they showed an almost matching lower bound for sketching submodular functions with polynomially many value queries. Another lower bound is implied by the work of Balcan and Harvey [3]. They showed that there are certain matroid rank functions, a subclass of submodular valuations, for which *every* sketch fails to provide an approximation ratio better than $n^{\frac{1}{3}}$. Notice that this bound is unconditional in the sense that it holds even if we have unlimited computational power.

Can we obtain good sketches for the more general class of subadditive functions? Subadditive functions are significantly "harder to handle" than submodular functions. For example, Dobzinski et al. [9] showed that there is no polynomial time $O(1)$-approximation for the problem of maximizing subadditive functions subject to a cardinality constraint by value queries, whereas the classical greedy algorithm [20] gives a $\frac{e}{e-1}$-approximation for submodular functions. As another example, in Appendix 6 we show that there are simple subadditive functions for which no submodular function provides a better than $n^{1/2}$-approximation. Given subadditive functions' looser structures, and the fact that [13] used substantial techniques specific to submodular functions, it is unclear whether one can obtain good sketches for subadditive functions.

Our first main result shows that for every subadditive function there exists a $\tilde{O}(\sqrt{n})$-sketch that can be found with only polynomially many queries, albeit with a form of queries called *demand queries*, which are more powerful than value queries. Demand queries are motivated in economic settings where an agent with a certain valuation function facing a set of items, each with a price tag, is required to report a subset of items that maximizes his utility. In mathematical terms, given a function $f : 2^N \to \mathbb{R}_+$, a demand query on f presents a price vector $p \in \mathbb{R}_+^N$ and gets as an answer a bundle $S \in \arg\max_{T \subseteq N}\{f(T) - \sum_{j \in T} p_j\}$. Demand queries have been broadly used and studied in the literature of algorithmic game theory (see, for example, [7]), and are known to be strictly more powerful than value queries in the sense that one can simulate a value query by polynomially many demand queries, but the converse is not true [8].

We prove that the last result is essentially the best one can hope for. First, we show that the approximation ratio is essentially tight in the sense that there are XOS functions (a subclass of subadditive functions) that do not admit an $o(\sqrt{n})$-sketch. Second, we prove that value queries are not powerful enough to obtain good sketches: a deterministic algorithm that always finds $O(n^{1-\epsilon})$-sketches must use superpolynomially many value queries, for any $\epsilon > 0$. This shows that one must use stronger queries in order to obtain good sketches.

Whereas for subadditive functions we show that there are efficient algorithms using demand queries that always produce sketches whose size matches the information theoretic bound, we show that this is not always the case for other valuation classes. We consider the class of OXS functions, a subclass of submodular functions that admits a 1-sketch [6] (i.e., the function can be fully described in polynomial space). However, we show that obtaining an $n^{\frac{1}{2}-\epsilon}$-sketch requires exponentially many value queries. Moreover, since OXS functions belong to the class of gross substitutes valuations for which a *demand* query can be simulated by polynomially many value queries, we have that obtaining an $n^{\frac{1}{2}-\epsilon}$ sketch requires exponentially many demand queries.

We then consider another well-studied subclass of submodular functions, *coverage functions*. We show that coverage valuations admit short sketches of arbitrary precision. A coverage function f is defined on a set N, each element of which corresponds to a subset of some universe Ω of points with non-negative weights, and the value of f on $S \subseteq N$ is the weight of the points in Ω covered by the sets corresponding to elements of S. We show that by appropriately sampling and reweighting points from the universe one can obtain a $(1+\epsilon)$-sketch which can be described in $poly(n, \frac{1}{\epsilon})$ space. It is an open question to obtain this sketch with either value queries or demand queries.

Finally we show an interesting connection between sketching and learning. Balcan and Harvey [3] introduce the problem of learning submodular functions: given bundles $S_1, ..., S_{\text{poly}(n)}$ sampled i.i.d. from some distribution D and their values $f(S_1), ..., f(S_{\text{poly}(n)})$ according to some unknown function f, can we find an almost correct sketch of f for subsequent samples drawn from D? They coin the term *PMAC learning* (**p**robably **m**ostly **a**pproximately **c**orrect) to refer to this type of

[1]For simplicity we assume that for each set S, $f(S) \in \{0, 1, 2, ..., poly(n)\}$. All results can be generalized to the case where f takes arbitrary real values.

approximation guarantee, in which the sketch may fail to be an α-approximation for certain bundles, but with high probability a subsequent sample from D will not belong to this exceptional set of bundles. We show that PMAC learning can be done *for every class of valuations*: if a class of valuations admits an α-sketch, then it can be learned. Using the results in the paper this implies that a $\tilde{O}(\sqrt{n})$-sketch of subadditive functions can be learned, as well as arbitrarily good sketches for coverage and OXS valuations. However, the bounds we prove are only information theoretic ones and we do not show the existence of a computationally efficient learning algorithm for this problem.

Independently of our work, Balcan et al. [2] obtained related, but largely complementary, results on learning valuation functions. They give computationally efficient algorithms for PMAC learning a $\tilde{O}(\sqrt{n})$-sketch of a subadditive function using polynomially many value queries. In comparison, our sketching algorithm satisfies a stronger (i.e. pointwise) approximation guarantee but uses demand queries, and therefore is not computationally efficient in general. Balcan et al. [2] also present improved PMAC guarantees for certain classes of functions, such as XOS functions represented by a polynomial number of clauses, and they prove hardness results for learning XOS and OXS valuations that are analogous to the sketching lower bounds we present here.

1.1 Sketching Subadditive Valuations: a Brief Overview

Let us sketch why every submodular function has an $O(\sqrt{n})$-sketch [13]. Every submodular function f defines a polymatroid $P_f : \{x \in \mathbb{R}_+^{|S|} \mid x(T) \leq f(T), \forall T \subseteq S\}$, where $x(T) = \sum_{j \in T} x_j$. The basic idea is to show the existence of an ellipsoid E such that $\frac{1}{\sqrt{n}} P_f \subseteq E \subseteq P_f$. If we have such ellipsoid E we can use it as our sketch, since for every S we have that $f(S) = \max\{(1_S)^T x | x \in P_f\}$. Towards this end, we consider the symmetrized version of P_f, $\hat{P}_f \triangleq \{x \in \mathbb{R}^n \mid |x| \in P_f\}$. The renowned John's theorem states that for any centrally symmetric convex body P in $\mathbb{R}^n$, there exists an ellipsoid E such that $\frac{1}{\sqrt{n}} P \subseteq E \subseteq P$. We get our sketch by applying John's theorem to P_f. Notice that this proves the *existence* of an $O(\sqrt{n})$-sketch, but not how to efficiently find it.

We now want to show that every *subadditive* function has an $\tilde{O}(\sqrt{n})$-sketch. As a first step, we show that every *XOS* function (a.k.a. as *fractionally subadditive*) has an $O(\sqrt{n})$-sketch. A function is XOS if there exists additive valuations[2] $a_1, \ldots, a_t$ such that $v(S) = \max_i a_i(S)$. It is known that XOS functions are a proper superclass of submodular functions and a proper subclass of subadditive functions. The key observation here is that the class of XOS functions is exactly the class for which $f(S) = \max\{(1_S)^T x | x \in P_f\}$, for every S, by taking x_j to be $a_{i^*}(\{j\})$ for all $j \in S$, where $i^* = \arg\max_i a_i(S)$. Hence every XOS function admits an $O(\sqrt{n})$-sketch.

Now we extend this result to subadditive functions. We use a result in [10] that shows that for every subadditive function there exists an XOS function that $O(\log n)$-approximates it. This shows that every *subadditive* function has an $O(\log n \sqrt{n})$-sketch: take the XOS function that $O(\log n)$ approximates it, and provide the $O(\sqrt{n})$-sketch that was obtained using the ellipsoidal approach. We are left with showing that such a sketch can indeed obtained algorithmically.

A crucial insight of [13] is that the problem can be reduced to the problem of finding a point in P_f that is "far" from a given ellipsoid, which in turn is equivalent to the following optimization problem for any $c \in \mathbb{R}_+^n$:

$$\max \sum_i c_i^2 x_i^2 \quad \text{s.t. } x \in P_f$$

where c_i's are coefficients specifying an ellipsoid $\sum_i c_i^2 x_i^2 \leq 1$. A β-approximation for this problem will give a $\sqrt{\beta n}$-approximation for P_f, and hence a $O(\sqrt{\beta n} \log n)$-approximation for a subadditive f.

For submodular functions, Goemans et al. showed that: (1) under certain conditions, a "scaled" polymatroid (corresponding to heterogeneous c_i's) can be approximated by an "unscaled" polymatroid (corresponding to the same c_i's) within an $O(\log n)$ factor, and (2) the classical greedy algorithm for maximizing submodular functions subject to a cardinality constraint [20] provides a $(1 - \frac{1}{e})^2$ approximation for the unscaled case. Notably, (2) requires only value queries. Unfortunately, the approach of [13] fails in the case of subadditive functions.

Therefore we develop new machinery to handle subadditive functions. En route, we significantly simplify the first step for more general polytopes while avoiding proving a structural theorem a la [13]. We start by observing that for any elements i, j in S, if $f(\{i\})$ is significantly larger than $f(\{j\})$, then for any $T \subseteq S$ that contains both i and j, the value of $f(T)$ will not change much if we ignore the contribution of j, a consequence of subadditivity. Our goal now is to provide a set of ellipsoids that will approximate f in different magnitudes of values. This enables us to reduce the problem to the still-challenging problem of approximating the quadratic program where all c_i's are equal. In

[2] A valuation v is additive if for every set S we have that $v(S) = \Sigma_{j \in S} v(\{j\})$.

addressing this, we discover an interesting substructure for subadditive functions, which we call *universal sequences.*

Definition 1.1. *Let* $f : 2^N \to \mathbb{R}_+$ *be a function. A sequence* $\emptyset = T_0 \subsetneq T_1 \subsetneq \cdots \subsetneq T_n = N$ *is called a* γ-universal sequence *of* f *if, for any set* S *we have that* $f(S) \le \gamma \cdot f(T_{|S|})$.

Note that each T_{i+1} has one more element than T_i. In plain words, a universal sequence is an ordering of the elements in N such that for any $k \le n$, the first k elements in this ordering provide a γ-approximation for the maximization problem subject to cardinality constraint of k. The greedy algorithm for submodular functions, for example, shows that there is a $\frac{e}{e-1}$-universal sequence for any *submodular* function. We show that any subadditive function admits a 4-universal sequence. Then we construct a vector in P_f using such a sequence and show that the vector is an $O(\log^2 n)$-approximation for the quadratic program by exploiting the symmetry and convexity of the objective function.

2 Approximating Subadditive Functions using Demand Queries

In this section we describe an algorithm that outputs an $\tilde{O}(\sqrt{n})$ approximation to any subadditive function $f : 2^{[n]} \to \mathbb{R}_+$ using polynomially many demand queries.

We follow the approach of ellipsoidal approximation introduced in [13]. An ellipsoid $E_A \subseteq \mathbb{R}^n$ defines a function L by mapping each $S \subseteq [n]$ to $\max_{x \in E_A} 1_S^T x$, where 1_S is the indicator vector for S. In the following we often use the term ellipsoid to also refer to the function it defines. Recall that a function $f : 2^{[n]} \to \mathbb{R}_+$ defines a polytope $P_f = \{x(S) \le f(S), \forall S \subseteq [n] \mid x \in \mathbb{R}^n_+\}$, where $x(S) = \sum_{i \in S} x_i$. In [13] the following lemma is proven:

Lemma 2.1. ([13]) *Let* f *be a function and* P_f *be the polytope it defines. If we have an algorithm* A *that provides a* β*-approximation for the quadratic program*

$$\max \sum_i c_i^2 x_i^2 \quad \text{s.t. } x \in P_f$$

for every i, $1 \le c_i f(\{i\}) \le \sqrt{n+1}$, *then we can find an ellipsoid that provides a* $O(\sqrt{n\beta})$*-approximation in time polynomial in the running time of* A.

Unlike [13] we do not show how to solve this quadratic program for all c_i's. Nevertheless, we show that to obtain an $\tilde{O}(\sqrt{n})$-sketch for subadditive functions it suffices to be able to solve the case when all c_i's are 1.

Lemma 2.2. *Let* f *be a subadditive function and* P_f *be the polytope it defines. If we have an algorithm* A *that provides a* β*-approximation for the quadratic program*

$$\max \sum_i x_i^2 \quad \text{s.t. } x \in P_f$$

then we can find an $O(\sqrt{n\beta}\log n)$*-sketch for* f *in time polynomial in the running time of* A.

Proof. We will reduce the problem in Lemma 2.1 to the current problem with a loss of a $\log n$ factor in the solution. Without loss of generality, we assume the items in N are ordered such that $f(\{1\}) \ge f(\{2\}) \ge \ldots \ge f(\{n\})$.

Definition 2.1. *For each element* $i \in [n]$, *define its* vicinity *to be* $V_i = \{j \mid i \le j \le n, nf(j) \ge f(i)\}$.

For a subset $S \subseteq [n]$, let $f|_S$ be $f|_S(T) = f(T \cap S)$ for any $T \subseteq [n]$. It is easy to see that $f|_{V_i}$ is still a subadditive function. The algorithm will compute n ellipsoids such that each ellipsoid is an $\tilde{O}(\sqrt{n})$ approximation for $f|_{V_i}$, respectively. For a subset $S = \{i_1, i_2, \ldots, i_m \mid i_1 < i_2 < \ldots < i_m\}$, we use the ellipsoid corresponding to $f|_{V_{i_1}}$ to approximate $f(S)$. If the corresponding ellipsoid is an $\tilde{O}(\sqrt{n})$ approximation for $f|_{V_{i_1}}$, then since

$$f(S) \le f|_{V_{i_1}}(S) + n \cdot n^{-1} f(i_1) \le 2f|_{V_{i_1}}(S),$$

we will have obtained an $\tilde{O}(\sqrt{n})$ approximation for $f(S)$ itself. From this point on we will assume that $f(\{1\}) \ge f(\{2\}) \ge \ldots \ge f(\{n\}) \ge \frac{f(\{1\})}{n}$.

Recall that $1 \le c_i f(\{i\}) \le \sqrt{n+1}$. By the assumption that $f(\{1\}) \ge \ldots \ge f(\{n\}) \ge \frac{f(\{1\})}{n}$, the ratio $\frac{\max_i c_i}{\min_j c_j}$ is polynomially bounded. We reduce the problem by first rounding each c_i down to the largest power of 2, and thereby grouping them into $O(\log n)$ bins $B_1, \ldots, B_k$; then we solve the optimization problem

$$\max_{i \in B_i} \sum x_i^2 \quad \text{s.t. } x \in P_f$$

for each B_i. It is easy to see that the best solution $x \in P_f$ obtained will be an $O(\log^2 n)$ approximation for the problem with different c_i's.

Hence, it is left to show that one can approximate the optimization problem

$$\max \sum_i x_i^2 \quad \text{s.t. } x \in P_f$$

within an $O(\log^4 n)$ factor using polynomially demand queries. We first introduce the notion of *universal sequences* both for the ease of presentation and for its own interest:

DEFINITION 2.2. *Let $f : 2^N \to \mathbb{R}_+$ be a function. A sequence $\emptyset = T_0 \subsetneq T_1 \subsetneq \ldots \subsetneq T_n = N$ is called a γ-*universal sequence *of f if, for any set S we have that $f(S) \leq \gamma \cdot f(T_{|S|})$.*

In particular, for submodular functions the greedy algorithm produces a $\frac{e}{e-1}$-universal sequence. For subadditive functions we show that there exists a 4-universal sequences. This will be a by product of the following algorithm that finds a poly logarithmic approximation to the quadratic program. The algorithm makes use of the following notion:

DEFINITION 2.3. ([11, 10]) *Let $f : 2^N \to \mathbb{R}_+$ and $S \subseteq N$. For each $j \in S$, let p_j be a non-negative real number. We say that the p_j's are α-*supporting prices *for S if $\Sigma_j p_j \geq \alpha f(S)$ and $\forall T \subseteq S$, $\Sigma_{j \in T} p_j \leq f(T)$.*

In the algorithm we assume without loss of generality that n is a power of 2 – we can always add more items with zero value):

1. Let $T_0 = \emptyset$.
2. For each k. $k = 1, 2, \ldots, \log_2 n$:
   (a) Find a set S, $|S| = 2^k$ such that for every T, $|T| = 2^k$ we have that $\alpha \cdot f(S) \geq f(T)$.
   (b) Partition S to S_1 and S_2, $|S_1| = |S_2|$. Let $S_k = \arg\max_{S \in \{S_1, S_2\}}(f(S))$.
   (c) Let $T_k = T_{k-1} \cup S_k$.
   (d) Let $x_{|T_{k-1}|+1}, \ldots, x_{|T_k|}$ be $O(\log n)$-supporting prices of the set $T_k \setminus T_{k-1}$.
3. Let $\hat{x} = \frac{x}{2 \log n}$.

Step (2a) is the problem of optimizing a subadditive function subject to a cardinality constraint. A 2-approximation for this problem that uses only polynomially many demand queries was provided in [1] ($\alpha = 2$). Step (2d) can be implemented using the algorithm of [10] that finds $O(\log n)$ supporting prices using polynomially many demand queries[3]. Hence we have that the algorithm can be implemented using only polynomially many demand queries.

[3] It is known that for general subadditive functions, $O(\log n)$-supporting prices are the best that one can hope for; however, with XOS functions, for which 1-supporting prices exist, it remains open whether one can use polynomially many demand queries to get better than $O(\log n)$-supporting prices.

We first show that the algorithm finds a 4-universal sequence. This will later be used to show that x is an approximate solution to the quadratic problem.

PROPOSITION 2.1. *Let f be a subadditive function. The algorithm finds a 4α-universal sequence.*

Proof. The proof relies on the following claim:

CLAIM 2.1. *For every k, $k \leq \log_2 n$, and every T, $|T| = 2^k$, we have that $2\alpha \cdot f(T_k) \geq f(T)$.*

Proof. Let S be the bundle we obtained in the k'th iteration of Step 2a. Let T be some bundle of size 2^k. By subadditivity and the guaranteed approximation ratio α we have that $\alpha(f(S_1) + f(S_2)) \geq f(T)$. We took S_k to be the larger of the two, and hence $f(S_k) \geq \frac{1}{2\alpha} f(T)$. Now by construction $T_k \supseteq S_k$, and by monotonicity we have proved the claim.

By ordering the elements in each S_k arbitrarily and then add them one by one to a sequence of sets, we obtain a full sequence $\emptyset = U_0 \subsetneq U_1 \subsetneq U_2 \ldots \subsetneq U_n = N$. Note that every T_k occurs in this sequence. We can now prove the proposition: For any set $S \subseteq N$, we consider T_k where k is $\lfloor \log_2 |S| \rfloor$. From Claim 2.1 we have $f(T_k) \geq \frac{1}{2\alpha} \max_{T \subseteq N, |T| = 2^k} f(T) \geq \frac{1}{4\alpha} f(S)$, where the last inequality comes from subadditivity.

LEMMA 2.3. *$\hat{x}$ produced by the algorithm is an $O(\log^4 n)$ approximation for the quadratic program.*

The proof consists of the following claims.

CLAIM 2.2. *$\hat{x}$ is in P_f.*

Proof. By definition of P_f, it suffices to show that $\forall S \subseteq N$, $\hat{x}(S) \leq f(S)$. We note that

$$\begin{aligned} \hat{x}(S) &= \frac{x(S)}{2 \log n} \\ &\leq \max_i x(S \cap (T_i \setminus T_{i-1})) \\ &\leq \max_i f(S \cap (T_i \setminus T_{i-1})) \\ &\leq f(S) \end{aligned}$$

The first inequality is valid because there are only $(\lfloor \log_2 n \rfloor + 1)$ different S_i's, and the second inequality follows from the fact that x is defined as a supporting price of $f(T_i \setminus T_{i-1})$.

CLAIM 2.3. *$\forall k \leq \log_2 n$, $x(T_k) \geq \frac{f(T_k)}{O(\log n)}$.*

Proof. We prove this by induction. For $k = 1$, we have $x(T_1) = \frac{f(T_1)}{O(\log n)}$ by definition of supporting prices. Now suppose the claim is true for k, then by subadditivity

$$\begin{aligned} f(T_{k+1}) &\leq f(T_{k+1} \setminus T_k) + f(T_k) \\ &= O(\log n)(x(T_{k+1} \setminus T_k) + x(T_k)) \\ &= O(\log n)x(T_{k+1}) \end{aligned}$$

The equality follows from the definition of supporting prices and the induction hypothesis.

Now using $\hat{x}$, we define a symmetric[4] submodular function $g : 2^N \to \mathbb{R}$. For $S \subseteq N$, we let $g(S) = \max_{T \subseteq N, |T|=|S|} \hat{x}(S)$. It is easy to verify that g is indeed a submodular function. Let P_g be the polymatroid it defines, then we have

Claim 2.4. $P_g \supseteq \frac{P_f}{O(\log^2 n)}$. [5]

Proof. It suffices to show that for any $y \in P_f$ and $S \subseteq N$, $\frac{y(S)}{O(\log^2 n)} \leq g(S)$. Let k be $\lfloor \log_2 |S| \rfloor$, then by Claim 2.1 and Claim 2.3, we have

$$\begin{aligned} f(S) &\leq 4f(T_k) \\ &= O(\log n)x(T_k) \\ &= O(\log^2 n)\hat{x}(T_k) \\ &\leq O(\log^2 n)g(T_k) \\ &\leq O(\log^2 n)g(S) \end{aligned}$$

Since the vertices of P_g are permutations of the coordinates of $\hat{x}$, we know that $\hat{x}$ is an optimal solution to the optimization problem

$$\begin{aligned} &\max \sum x_i^2 \\ &\text{s.t. } x \in P_g \end{aligned}$$

By Claim 2.4 and Claim 2.2, we have that $\hat{x}$ is a feasible solution and gives an $O(\log^4 n)$ approximation for the optimization problem

$$\begin{aligned} &\max \sum x_i^2 \\ &\text{s.t. } x \in P_f \end{aligned}$$

3 Lower bounds

We have seen that there is a deterministic algorithm to compute a $\widetilde{O}(\sqrt{n})$-sketch of any subadditive function using demand queries. In this section, we show that this result is essentially the best possible, in two respects. First, for any fixed $\varepsilon > 0$, it is not the case that every subadditive function admits a $O(n^{1/2-\varepsilon})$-sketch of polynomial size. In fact our bound even holds for XOS valuations. Previously, Balcan and Harvey [3] showed that every polynomial-size sketch of submodular functions cannot have an approximation ratio better than $n^{\frac{1}{3}}$. Second, for deterministic algorithms that are limited to *value* queries, it is impossible to obtain a $O(n^{1-\varepsilon})$-sketch in polynomial query complexity.

We also consider the class of OXS functions. This class was defined in [16] and is equivalent to the to the class of weighted rank function of a transversal matroid (and hence is a subclass of the class of submodular valuations). This class can be represented in $O(n^2)$ space [6]. In contrast, we show that algorithmically obtaining such a sketch via queries is hard: an $O(n^{1/2-\varepsilon})$-sketch requires an exponential number of value queries. Moreover, for this class a demand query can be implemented via a polynomial number of demand queries [4]. Hence we have that an $O(n^{1/2-\varepsilon})$-sketch requires an exponential number of demand queries.

3.1 A Tight Lower Bound for Sketching XOS Valuations

We show that XOS functions do not admit $n^{\frac{1}{2}-\epsilon}$-sketches. This slightly improves over the result of [3] that showed there are no $n^{\frac{1}{3}-\epsilon}$-sketches if the valuation function is submodular (in fact, even gross substitutes).

Definition 3.1. *A family C of subsets of $\{1, ..., n\}$ is (h, ℓ)-good if*

1. *For each $S \in C$, $|S| \geq h$.*
2. *For each $S, T \in C$ and $S \neq T$, $|S \cap T| \leq \ell$.*

Lemma 3.1. *If some C is (h, ℓ)-good then approximating XOS valuations to within a factor of better than h/ℓ requires representation length of at least $|C|$.*

Proof. For every subset $D \subseteq C$ we define an XOS valuation: $v_D(S) = \max_{T \in D} |T \cap S|$. Now we claim that for every $D \neq D' \subseteq C$ there exists a subset S such that $v_D(S)/v_{D'}(S) \geq h/\ell$ or $v_{D'}(S)/v_D(S) \geq h/\ell$. This will be true exactly for $S \in (D - D') \cup (D' - D)$. The bound on sketching length is implied since any sketching scheme with approximation ratio better than h/ℓ cannot give any two D's the same sketch.

Lemma 3.2. *For $h = \frac{1}{2}\sqrt{n}$, for every ℓ, there exists a family that is (h, ℓ)-good of size $n^{\Omega(\ell)}$.*

Proof. Choose any prime p such that $n/4 \leq p^2 \leq n$, and identify a subset of N with the set $\mathbb{F}_p^2$, where $\mathbb{F}_p$ denotes the field of integers modulo p. For each univariate polynomial P of degree at most ℓ over $\mathbb{F}_p$, add the set $S_P = \{(x, P(x)) \mid x \in \mathbb{F}_p\}$ to the collection C. Each

[4] By symmetric we mean a function depending only on the cardinality.

[5] We note that we do not prove that $P_g \subseteq P_f$, and this is not generally true.

set in C has cardinality $p \geq \frac{1}{2}\sqrt{n}$, and the intersection of any two sets $S_P, S_Q \in C$ has cardinality at most ℓ because the equation $P(x) = Q(x)$ is satisfied by at most ℓ values of x. Finally, a polynomial of degree at most ℓ is uniquely determined by a sequence of $\ell + 1$ coefficients in $\mathbb{F}_p$, so the cardinality of C is $p^{\ell+1} = n^{\Omega(\ell)}$.

THEOREM 3.1. *Polynomial-size sketches cannot approximate XOS to within a factor of $o(\sqrt{n})$. Subexponential-size sketches cannot approximate XOS to within a factor better than $O(n^{1/2-\epsilon})$.*

3.2 Inapproximability of Subadditive Functions using Deterministic Value Queries The following theorem is a lower bound for deterministic sketching of subadditive functions using value queries. The proof shows that polynomially many value queries cannot possibly provide enough evidence to distinguish the subadditive function $f(S) = |S|^{1-\delta}$ from a different subadditive function that takes the value n^δ on at least one set of size $n^{1-\delta}$.

THEOREM 3.2. *If a deterministic algorithm can compute an α-sketch of every subadditive function using $O(\text{poly}(n))$ value queries, then $\alpha = \Omega(n^{1-\varepsilon})$ for every fixed $\varepsilon > 0$.*

To prove the theorem, we begin with the following characterization of subadditive set functions.

LEMMA 3.3. *Let $(A_1, c_1), \ldots, (A_k, c_k) \in 2^N \times \mathbb{R}_+$ be a sequence of pairs consisting of a subset of N and a non-negative cost for that subset. Suppose that $N = \bigcup_{i=1}^k A_i$. The set function*

$$g(U) = \min \left\{ \sum_{j \in J} c_j \,\middle|\, \bigcup_{j \in J} A_j \supseteq U \right\},$$

called the min-cost-cover function *of $\{(A_i, c_i)\}_{i=1}^k$, is a non-negative, monotone, subadditive set function. Furthermore, every non-negative, monotone, subadditive set function can be represented as the min-cost-cover function of a suitably defined sequence $\{(A_i, c_i)\}_{i=1}^q$.*

Proof. Non-negativity and monotonicity of g are clear from the definition. Subadditivity follows from the observation that if $\bigcup_{j \in J} A_j \supseteq U$ and $\bigcup_{j \in J'} A_j \supseteq U'$ then $\bigcup_{j \in J \cup J'} A_j \supseteq U \cup U'$ while $c(J \cup J') \leq c(J) + c(J')$. (Here we are using the notation $c(J)$ to denote $\sum_{j \in J} c_j$.)

Conversely, if f is non-negative, monotone, and subadditive, then f is equal to the min-cost-cover function of the set of ordered pairs $(A, f(A))$ where A ranges over all subsets of N. The min-cost-cover of U is less than or equal to $f(U)$ because the set U covers itself, and it is not strictly less because f satisfies monotonicity and subadditivity.

We will also make use of the following form of the Chernoff bound.

LEMMA 3.4. *Let T be a random subset of an n-element set N, obtained by selecting every element independently with probability p, and let U be any other, fixed, subset of N. For any $1 \leq \ell \leq n$,*

$$\Pr\left(|T \cap U| \geq \max\{\ell, 2p|U|\}\right) < e^{-\ell/12}.$$

We are ready now to proceed with the proof of Theorem 3.2.

Proof. [Theorem 3.2] Let $\delta = \varepsilon/3$, and assume without loss of generality that $\delta < 1/2$. Consider the sequence of queries and responses that take place when we run the algorithm using the subadditive function $f(S) = |S|^{1-\delta}$. Denote this sequence by $(S_1, v_1), \ldots, (S_q, v_q)$, where $v_i = |S_i|^{1-\delta}$ for $1 \leq i \leq q$. We will construct a subadditive function g such that $g(S_i) = v_i$ for all $i \in \{1, \ldots, q\}$, but $g(T) \leq n^\delta$ for some other set T of cardinality at least $k = \lceil n^{1-\delta} \rceil$. When the algorithm observes the sequence of queries and responses $(S_1, v_1), \ldots, (S_q, v_q)$, its output f' must be an α-sketch of both f and g, hence $f(T)/\alpha \leq f'(T) \leq g(T)$. This implies that $\alpha \geq f(T)/g(T) \geq n^{(1-\delta)^2 - \delta} > n^{1-\varepsilon}$.

To construct the function g we use the probabilistic method. Let T be a random subset of N obtained by sampling each element independently with probability $p = 2k/n$, and let g be the min-cost-cover function of the sequence $(S_1, v_1), (S_2, v_2), \ldots, (S_q, v_q), (T, n^\delta)$. Chebyshev's inequality implies that with probability at least $\frac{1}{2}$, $|T| \geq 2k - \sqrt{n} \geq k$. We will complete the proof by showing that $\Pr(\exists i \text{ s.t. } g(S_i) \neq f(S_i))$ is less than $\frac{1}{2}$ for large enough n. In fact, we will show that the event $\{\exists i \text{ s.t. } g(S_i) \neq f(S_i)\}$ is contained in the union of the events $\{|T \cap (S \setminus S')| \geq \max\{n^\delta, p|S \setminus S'|\}\}$, where S, S' range over all pairs of sets in the collection $\{\emptyset, S_1, S_2, \ldots, S_q\}$. By the union bound and Lemma 3.4, the probability that there exists i such that $g(S_i) \neq f(S_i)$ is at most $O(q^2)\exp(-n^\delta/12)$ and this is less than $\frac{1}{2}$ for sufficiently large n.

To finish the proof, we must show that the assumption that $g(S_i) \neq f(S_i) = |S_i|^{1-\delta}$ implies the existence of two sets S, S' in the collection $\{\emptyset, S_1, S_2, \ldots, S_q\}$ such that

$$|T \cap (S \setminus S')| \geq \max\{n^\delta, 2p|S \setminus S'|\}. \tag{3.1}$$

Clearly, by construction, $g(S_i) \leq |S_i|^{1-\delta}$, so assume that the inequality is strict. It means that there is an index set $J \subseteq [q]$ such that either

$$c(J) < |S_i|^{1-\delta} \text{ and } \bigcup_{j\in J} S_j \supseteq S_i,$$

or

$$n^\delta + c(J) < |S_i|^{1-\delta} \text{ and } T \cup \left(\bigcup_{j\in J} S_j\right) \supseteq S_i.$$

The first alternative is not possible, since the set function $f(S) = |S|^{1-\delta}$ is subadditive. So assume the second alternative holds. Letting $V = (T \cap S_i) \setminus \left(\bigcup_{j\in J} S_j\right)$, we have $S_i \subseteq V \cup \left(\bigcup_{j\in J} S_j\right)$, and by the subadditivity of $f(S) = |S|^{1-\delta}$ this implies that

$$|V|^{1-\delta} \geq |S_i|^{1-\delta} - \sum_{j\in J} |S_j|^{1-\delta} > n^\delta. \tag{3.2}$$

The set $S_i \setminus V$ is contained in $\bigcup_{j\in J} S_j$. Partition $S_i \setminus V$ arbitrarily into disjoint subsets $\{W_j\}_{j\in J}$ such that $W_j \subseteq S_j$ for all $j \in J$. For each $x \in S_i$ define

$$w(x) = \begin{cases} |W_j|^{-\delta} & \text{if } x \in W_j \text{ for some } j \in J \\ 0 & \text{if } x \in V. \end{cases}$$

Note that the cases are mutually exclusive and exhaustive, so $w(x)$ is well-defined for all $x \in S_i$. We have

$$\sum_{x\in S_i} w(x) = \sum_{j\in J} |W_j| \cdot |W_j|^{-\delta} \leq \sum_{j\in J} |S_j|^{1-\delta} = c(J) < |S_i|^{1-\delta}, \tag{3.3}$$

so

$$\frac{1}{|S_i|} \sum_{x\in S_i} w(x) < |S_i|^{-\delta}. \tag{3.4}$$

The argument now splits into two cases. If $|W_j| \leq \frac{1}{2}|S_i|$ for all $j \in J$ then we have $w(x) \geq 2^\delta |S_i|^{-\delta}$ for all $x \in S_i \setminus V$, hence, by (3.3),

$$2^\delta |S_i|^{-\delta} \cdot |S_i \setminus V| < |S_i|^{1-\delta}$$
$$|S_i \setminus V| < 2^{-\delta}|S_i|$$
$$|V| > \left(1 - 2^{-\delta}\right)|S_i| > \frac{\delta}{2}|S_i|, \tag{3.5}$$

using the fact that $1 - 2^{-x} > x/2$ for all $0 < x < 1$. For sufficiently large n, the right side is greater than $2p|S_i|$, so combining (3.5) with (3.2), we obtain $|V| \geq \max\{n^\delta, 2p|S_i|\}$, which confirms (3.1) with $S = S_i, S' = \emptyset$, since $T \cap S_i \supseteq V$.

The remaining case is that $|W_j| > \frac{1}{2}|S_i|$ for some $j \in J$. In that case, the average value of $w(x)$ over $x \in W_j$ is $|W_j|^{-\delta}$, which is greater than $|S_i|^{-\delta}$, whereas the average value of $w(x)$ over $x \in S_i$ is strictly less than $|S_i|^{-\delta}$ by (3.4). Consequently the average value of $w(x)$ over $x \in S_i \setminus W_j$ must be strictly less than $|S_i|^{-\delta}$. Whenever $x \in S_i \setminus W_j$ and $w(x) > 0$, then x belongs to some $W_{j'}$ such that $|W_{j'}| < \frac{1}{2}|S_i|$, and consequently $w(x) > 2^\delta |S_i|^{-\delta}$. Arguing as in case 1, this implies that

$$|V| > \frac{\delta}{2}|S_i \setminus W_j| \geq \frac{\delta}{2}|S_i \setminus S_j|. \tag{3.6}$$

Combining (3.6) with (3.2), we obtain $|V| \geq \max\{n^\delta, 2p|S_i \setminus S_j|\}$, which confirms (3.1) with $S = S_i, S' = S_j$, since $T \cap (S_i \setminus S_j) \supseteq V$.

3.3 A Lower Bound for OXS Functions

THEOREM 3.3. *Let f be an OXS valuation and let A be an algorithm that that provides an $n^{\frac{1}{2}-\epsilon}$-sketch, using value queries and demand queries. A does not make a polynomial number of such queries.*

Proof. Since demand queries for OXS valuations can be simulated using a polynomial number of value queries, we assume henceforth that A makes only value queries.

Start with the complete bipartite graph $K_{k,n}$ where $k = \delta n$. Pick a random subset B of δn nodes on the RHS, and an arbitrary subset A of $\epsilon\delta n$ nodes on the LHS. (We will fix ϵ, δ later to be $\Theta(\sqrt{(\log n/n)}$.) Delete all edges between B and A^c (where A^c is the intersection of the complement of A and the vertices in the LHS). For a subset S of nodes on the RHS, let $v(S)$ denote the maximal matching size that only matches RHS nodes in S. Observe that we can write

$$v(S) = \min\{\delta n, |S \cap B^c| + \min\{\epsilon\delta n, |S \cap B|\}\}$$

(Since nodes in B can only contribute $\epsilon\delta n$, and because there are only δn LHS nodes.) Notice that v is a rank function of a matching matroid, hence it is indeed an OXS valuation.

It is enough to prove that, for every S, with high probability over the choice of B, $v(S) = \min\{|S|, \delta n\}$. (Then you learn nothing about B from any of your value queries.) This will show that we cannot distinguish with polynomially many value queries between $v(B) = \delta n$ and $v(B) = \epsilon\delta n$, and an approximation lower bound of $1/\epsilon$ will therefore follow.

Assume we choose ϵ, δ so that $\epsilon\delta n = \Omega(\log n)$. If $|S| = O(\epsilon\delta n)$ then $v(S) = |S|$. Otherwise, by Chernoff we have $|S \cap B|$ and $|S \cap B^c|$ concentrated around their expectations of $\delta|S|$ and $(1-\delta)|S|$, respectively. If $|S| = O(\epsilon n)$ then $|S \cap B| = O(\epsilon\delta n)$ and $v(S) = \min\{|S|, \delta n\}$. If $|S| = \Omega(\epsilon n)$ then $|S \cap B^c| = \Omega((1-\delta)\epsilon n)$, which is $\Omega(\delta n)$ provided $\epsilon \geq \delta/(1-\delta)$. In this case, $v(S) = \delta n$.

Finally, we choose $\epsilon = 2\delta = \Theta(\sqrt{\log n/n})$ and get a lower bound of $\Omega(\sqrt{n/\log n})$.

4 Coverage Functions Admit $(1+\epsilon)$-Sketches

A set function $f : 2^N \to \mathbb{R}_+$ is called a *coverage function* if there exists a finite set Ω, a weight function $w : \Omega \to \mathbb{R}_+$, and a function $g : N \to 2^\Omega$ such that $f(S) = w\left(\bigcup_{i \in S} g(i)\right)$ for all $S \subseteq N$.

We will present a sampling algorithm which produces a coverage function $\hat{f}$ on a set of points Ω' with $|\Omega'| \le \frac{27\,n^2}{\epsilon^2}$, such that $\hat{f}$ is a $(1+\epsilon)$-sketch of f with high probability. We will assume without loss of generality that $0 < \varepsilon < 1$ and that $\Omega = \bigcup_{i \in N} g(i)$.

Algorithm: Let $t = \frac{27\,n^2}{\varepsilon^2}$. Define $q(x) = \max\left\{ \frac{w(x)}{f(\{i\})} \,\middle|\, i \in N, x \in g(i)\right\}$ for $x \in \Omega$. Let $p(x) = \frac{q(x)}{\sum_{y\in\Omega} q(y)}$. The algorithm draws t independent random samples $x_1, \ldots, x_t$ from the distribution on Ω specified by p. It defines $\Omega' = \{x_1, \ldots, x_t\}$ and sets $g'(i) = \Omega' \cap g(i)$ for all $i \in N$. To define the weight of an element $x \in \Omega'$, we let $m(x)$ denote the number of times x occurs in the sequence and set $w'(x) = m(x) \cdot \frac{w(x)}{t \cdot p(x)}$. Finally, define $\hat{f}$ to be the coverage function specified by Ω', w', and g'. The algorithm outputs the function $\left(1 + \frac{\varepsilon}{3}\right)^{-1} \hat{f}$.

4.1 Proof of $(1+\varepsilon)$-approximation

We will prove that this algorithm outputs a $(1+\varepsilon)$-sketch of f. We need the following simple form of the Chernoff bound, which is well known [19] in the special case $m = \mu$, and whose general case follows from that special case by a trivial scaling argument.

LEMMA 4.1. *Suppose $Y_1, \ldots, Y_t$ are mutually independent random variables satisfying $\mathbb{E}\left[\sum_{i=1}^t Y_i\right] = \mu$, and suppose that each of the variables Y_i is supported in the interval $[0, \mu/m]$ for some $m > 0$. Then for $0 < \delta < 1$, we have*

$$\Pr\left((1-\delta)\mu \le \sum_{i=1}^t Y_i \le (1+\delta)\mu\right) > 1 - 2e^{-(\delta^2/3)m}.$$

Remark. The multiplicative form of the Chernoff bound is usually stated as two separate bounds, one for $\Pr((1-\delta)\mu > \sum Y_i)$ and another for $\Pr(\sum Y_i > (1+\delta)\mu)$. The version stated above follows by summing these two bounds and subtracting from 1, then using the fact that $e^{\delta-(1+\delta)\ln(1+\delta)} > e^{-\delta^2/3}$ for $0 < \delta < 1$.

Fix an arbitrary set $S \subseteq N$, let $U = \bigcup_{i\in S} g(i)$, and define random variables $Y_1, \ldots, Y_t$ by

$$Y_i = \begin{cases} \frac{w(x_i)}{t \cdot p(x_i)} & \text{if } x_i \in U \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\hat{f}(S) = \sum_{i=1}^t Y_i$, and that the random variables $Y_1, \ldots, Y_t$ are mutually independent. To apply the Chernoff bound, we will need the following pair of claims.

CLAIM 4.1. *For every $S \subseteq N$, $\mathbb{E}[\hat{f}(S)] = f(S)$.*

Proof. For each i we have

$$\begin{aligned} \mathbb{E}[Y_i] &= \sum_{x\in U}\left(p(x) \cdot \frac{w(x)}{t \cdot p(x)}\right) \\ &= \sum_{x \in U} \frac{w(x)}{t} \\ &= \frac{1}{t} w(U) \\ &= \frac{1}{t} f(S) \end{aligned}$$

The claim follows by summing over $i = 1, \ldots, t$.

CLAIM 4.2. *For every $x \in \Omega$ and every $S \subseteq N$, if $x \in \bigcup_{i\in S} g(i)$ then $\frac{w(x)}{t \cdot p(x)} \le \frac{\varepsilon^2}{27\,n} f(S)$.*

Proof. The key observation is that $\sum_{y\in\Omega} q(y) \le n$. To prove this, define for each $i \in N$ a set $h(i) = \left\{y \in g(i) \mid q(y) = \frac{w(y)}{f(\{i\})}\right\}$. Note that every $y \in \Omega$ belongs to at least one of the sets $h(i)$. Therefore

$$\begin{aligned} \sum_{y\in\Omega} q(y) &\le \sum_{i \in N} \sum_{y \in h(i)} q(y) \\ &= \sum_{i\in N} \frac{\sum_{y\in h(i)} w(y)}{f(\{i\})} \\ &= \sum_{i\in N} \frac{\sum_{y\in h(i)} w(y)}{\sum_{y \in g(i)} w(y)} \\ &\le n \end{aligned}$$

Now, for any $x \in \bigcup_{i\in S} g(i)$, we have

$$\begin{aligned} \frac{w(x)}{t \cdot p(x)} &= \frac{\varepsilon^2 w(x)}{27\,n^2} \cdot \frac{\sum_{y\in\Omega} q(y)}{q(x)} \\ &\le \frac{\varepsilon^2}{27\,n} \cdot \frac{w(x)}{q(x)} \\ &= \frac{\varepsilon^2}{27\,n} \cdot \min\{f(\{i\}) \mid i \in N, x \in g(i)\} \\ &\le \frac{\varepsilon^2}{27\,n} f(S) \end{aligned}$$

Combining Claims 4.1 and 4.2 with Lemma 4.1 and using $\delta = \varepsilon/3$, $m = \frac{27\,n}{\varepsilon^2}$ in that lemma, we obtain the result that

$$\begin{aligned} \Pr\left(|\hat{f}(S) - f(S)| \le \frac{\varepsilon}{3} f(S)\right) &> 1 - 2e^{-(\varepsilon^2/27)(27\,n/\varepsilon^2)} \\ &= 1 - 2e^{-n} \end{aligned}$$

Taking the union bound over all sets $S \subseteq N$, we now see that with probability at least $1 - 2^{n+1}e^{-n}$, the function $\left(1+\frac{\varepsilon}{3}\right)^{-1}\hat{f}$ is an α-sketch of f where $\alpha = \left(1+\frac{\varepsilon}{3}\right)/\left(1-\frac{\varepsilon}{3}\right) < 1+\varepsilon$, where the last inequality follows from our assumption that $\varepsilon < 1$.

5 Learning via Sketching

Balcan and Harvey [3] introduce the problem of learning submodular functions. We are given *poly*(n) bundles that are drawn i.i.d. from some distribution and their values according to some unknown submodular function f. The goal is to output a "sketch" of the functions, i.e., its "learned" valuation.

We will show that if a class of valuations admits an α-sketch[6], then there exists a learning algorithm for this class. The caveat is that we only focus at the number of allowed queries and the information that can be learned from this. In particular we completely ignore the actual algorithmic challenges and assume a computationally unbounded learner whose only constraints are the access to the valuation function. This is in complete contrast to most work in learning theory that focuses on the algorithmic issues and for which the informational questions are trivial.

We stress while sketching may require arbitrarily complicated and numerous queries, we show that learning can always be done by value queries — as long as we allow some probability of error.

Definition 5.1. (Balcan and Harvey [3]) *A learning algorithm sees a sequence of labeled examples $(S_i, v(S_i))$ where S_i is drawn according to distribution D on subsets of N, and then is asked a query S drawn according to the same D and needs to output an approximation $v(S)$. We say it is an α-approximating PMAC learning (with parameters ϵ, δ) for C if for every distribution D and every valuation $v \in C$ we have that with probability of at least $1-\delta$ (over the choice of $\{S_i\}$ from D), $Pr_S[v(S) \le v(S) \le \alpha v(S)] \ge 1-\epsilon$.*

Lemma 5.1. *If a class of valuations V can be α-approximately sketched with length $l(n)$ then it can be α-PMAC-learned from $O((l(n)+\log\delta^{-1})/\epsilon)$ samples.*

Proof. We use the usual principle in learning theory stating that compression implies prediction. Let $s : V \to \{0,1\}^*$ be the sketching function that takes a valuation as input and outputs an α-sketch of it (of size $l(n) = poly(n)$). For each possible value q of length $l(n,t)$ let us define $v_q(S) = \min_{v|s(v)=q} v(S)$, and thus for all v and all S we have that $v_{s(v)}(S) \le v(S) \le \alpha v_{s(v)}(S)$.

Let us use $r = O((l(n,t) + \log\delta^{-1})/\epsilon)$ samples $(S_i, v(S_i))$ and find some q such that $v_q(S_i) \le v(S_i) \le \alpha v_q(S_i)$ for all samples S_i. Such q must exist simply since $q = s(v)$ for the real v is such. (Notice that this may be algorithmically hard, but since we only count queries, it can be done in terms of information available to the mechanism after seeing the r samples.) We will use v_q as our learned valuation, i.e., on input S reply with $v_q(S)$.

Now let us calculate the probability that this v_q is not a proper approximation for v, i.e. that on at least ϵ-fraction of S's (weighted as in D) we do not get an α-approximation. Fix a single q for which we do not have the required approximation and let us calculate the probability that all for all S_i's we did get the required approximation: it is at most $(1-\epsilon)^r$. Now we use the union bound over all possible q's of length $l(n)$ to conclude that the probability that we output some bad q is at most $2^{l(n)}(1-\epsilon)^r$, which is bounded by δ using the choice of parameters we gave.

Using the upper bounds in this paper we have that:

Corollary 5.1. *The following statements are true:*

- *The class of subadditive valuations can be $\tilde{O}(\sqrt{n})$-PMAC-learned.*
- *The class of coverage valuations can be $(1+\epsilon)$-PMAC-learned.*
- *The class of OXS valuations can be $(1+\epsilon)$-PMAC-learned.*

6 A gap between XOS and Submodular Functions

In this section we give an example of XOS function for which no submodular function is an $O(\sqrt{n})$ approximation.

We define an XOS function $f : 2^S \to \mathbb{R}_+$, $|S| = n$, as follows. Partition S evenly into $\sqrt{n}$ subsets, $T_1, \cdots, T_{\sqrt{n}}$, each of size $\sqrt{n}$. Let f_i be an additive function on T_i: $f_i(T) = |T \cap S_i|$, $\forall T \subseteq S$. Then let f be the maximum of these functions, i.e., $f = \max_i f_i$. By definition, f is XOS.

Theorem 6.1. *If g is a submodular function such that $\frac{f(T)}{\alpha} \le g(T) \le f(T)$ for all $T \subseteq S$, then $\alpha \ge \frac{\sqrt{n}}{2}$.*

Proof. We construct a sequence of elements $x_1, \ldots, x_{\sqrt{n}} \in S$ inductively using g. For $x \in S$, let $\phi(x)$ be the i such that $x \in T_i$. Define

[6] We assume only existence of a sketch and do not care how it can be obtained nor with what type of queries.

$$x_1 = \operatorname*{argmax}_x \{g(\{x\}) \mid x \in S\}$$

Let $M_i = \cup_{j=1}^{i-1} T_{\phi(x_j)}$ and $G_i = \{x_1, \ldots, x_{i-1}\}$. Define

$$x_i = \operatorname*{argmax}_x \{g(x \mid G_i) \mid x \in S \backslash M_i\}, \quad i = 2, 3, \cdots, \sqrt{n}$$

where $g(x \mid T)$ is $g(\{x\} \cup T) - g(T)$, the marginal value of x given T. In words, $x_1, \ldots, x_{\sqrt{n}}$ are a sequence produced by a greedy algorithm that maximizes the marginal utility at each step, subject to the constraint that each element chosen is from a different subset in the partition.

By considering the value of f on these elements, we have

$$\begin{aligned} 1 &= f(\{x_1, \ldots, x_{\sqrt{n}}) \\ &\geq g(\{x_1, \ldots, x_{\sqrt{n}}) \\ &= g(\{x_1\}) + \sum_{i=2}^{\sqrt{n}} g(x_i | \{x_1, \ldots, x_{i-1}\}) \end{aligned}$$

By submodularity and the procedure we selected the sequence, the terms on the right hand side are in a nonincreasing order, and therefore $b \triangleq g(x_{\sqrt{n}} \mid \{x_1, \ldots, x_{\sqrt{n}-1}) \leq \frac{1}{\sqrt{n}}$. On the other hand,

$$\begin{aligned} \sqrt{n} &= f(\{x_1, \ldots, x_{\sqrt{n}-1}\} \cup T_{\sqrt{n}}) \\ &\leq \alpha g(\{x_1, \ldots, x_{\sqrt{n}-1}\} \cup T_{\sqrt{n}}) \\ &\leq \alpha(1 + (\sqrt{n} - 1)b) \leq 2\alpha \end{aligned}$$

The theorem immediately follows.

References

[1] Ashwinkumar Badanidiyuru, Shahar Dobzinski, and Sigal Oren. Optimization with demand oracles. *CoRR*, abs/1107.2869, 2011.

[2] Maria Florina Balcan, Florin Constantin, Satoru Iwata, and Lei Wang. Learning valuation functions. *CoRR*, abs/1108.5669, 2011.

[3] Maria-Florina Balcan and Nicholas J. A. Harvey. Learning submodular functions. In Lance Fortnow and Salil P. Vadhan, editors, *STOC*, pages 793–802. ACM, 2011.

[4] Alejandro Bertelsen. Substitutes valuations and m♮-concavity". M.Sc. Thesis, The Hebrew University of Jerusalem, 2005.

[5] Kshipra Bhawalkar and Tim Roughgarden. Welfare guarantees for combinatorial auctions with item bidding. In Dana Randall, editor, *SODA*, pages 700–709. SIAM, 2011.

[6] Liad Blumrosen. Information and communication in mechanism design. PhD Thesis, The Hebrew University of Jerusalem, 2006.

[7] Liad Blumrosen and Noam Nisan. Combinatorial auctions. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[8] Liad Blumrosen and Noam Nisan. On the computational power of demand queries. *SIAM J. Comput.*, 39(4):1372–1391, 2009.

[9] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. *Mathematics of Operations Research*, 35:1–13, 2010.

[10] Shahar Dobzinski. Two randomized mechanisms for combinatorial auctions. In *APPROX*, 2007.

[11] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, 2005.

[12] Uriel Feige. On maximizing welfare when utility functions are subadditive. *Siam Journal on Computing*, 39:41–50, 2006.

[13] Michel X. Goemans, Nicholas J. A. Harvey, Satoru Iwata, and Vahab S. Mirrokni. Approximating submodular functions everywhere. In Claire Mathieu, editor, *SODA*, pages 535–544. SIAM, 2009.

[14] A. Krause and C. Guestrin. Beyond convexity: Submodularity in machine learning, 2008. http://submodularity.org.

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

[16] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

[17] L. Lovász. Submodular functions and convexity. In *Mathemtical Programming: The State of The Art*, pages 235–257. Bonn, 1982.

[18] Elchanan Mossel and Sébastien Roch. Submodularity of influence in social networks: From local to global. *SIAM J. Comput.*, 39(6):2176–2188, 2010.

[19] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms.* Cambridge University Press, Cambridge, 1995.

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

[21] D.M. Topkis. *Supermodularity and Complementarity.* Princeton University Press, 1998.