Learning Valuation Functions

Maria Florina Balcan NINAMF@CC.GATECH.EDU
Georgia Institute of Technology, Atlanta, GA.

Florin Constantin FLORHARV@GMAIL.COM
A9.com, Palo Alto, CA.

Satoru Iwata IWATA@KURIMS.KYOTO-U.AC.JP
Kyoto University, Japan.

Lei Wang LEIWA@MICROSOFT.COM
Microsoft AdCenter, Bellevue, WA.

Editor: Shie Mannor, Nathan Srebro, Robert C. Williamson

Abstract

A core element of microeconomics and game theory is that consumers have valuation functions over bundles of goods and that these valuations functions drive their purchases. A common assumption is that these functions are subadditive meaning that the value given to a bundle is at most the sum of values on the individual items. In this paper, we provide nearly tight guarantees on the efficient learnability of subadditive valuations. We also provide nearly tight bounds for the subclass of XOS (fractionally subadditive) valuations, also widely used in the literature. We additionally leverage the structure of valuations in a number of interesting subclasses and obtain algorithms with stronger learning guarantees.

Keywords: learning, valuations, no complementarities, algorithmic game theory, economics

1. Introduction

A central problem in commerce is understanding one's customers. Whether to assign prices to goods, to decide how to bundle products, or to estimate how much inventory to carry, it is critical for a company to understand customers' preferences. In economics and (algorithmic) game theory, these preferences are typically modeled as valuations, or monotone set functions, over subsets of goods. It is usually assumed that consumers' valuations are known in advance, or that they are drawn from a known distribution. In practice, however, these valuations must be learned. For example, given past data of customer purchases of different bundles, a retailer would like to estimate how much a (typical) customer would be willing to pay for new packages of goods that become available. Companies may also conduct surveys querying customers about their valuations[1]. Motivated by such scenarios, in this paper we investigate the learnability of classes of functions widely used throughout economics and (algorithmic) game theory to model consumers' valuations (Nisan et al., 2007). We focus on subadditive valuations expressing "no complementarities" (the value of the union of two disjoint bundles is no more than the sum of the values on each bundle), and two of its subclasses commonly used, in decreasing order of generality, XOS (Bhawalkar and Roughgarden,

[1] See the site http://bit.ly/ls774D for an example of an airline asking customers for a "reasonable" price for in-flight Internet.

2011; Dobzinski et al., 2005, 2006; Feige, 2006; Lehmann et al., 2001) and OXS (Buchfuhrer et al., 2010a,b; Day and Raghavan, 2006; Singer, 2010).

To analyze these problems we use the PMAC learning model for approximate distributional learning introduced by Balcan and Harvey (2011). In this model, a learning algorithm is given a collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ of polynomially many labeled examples drawn i.i.d. from some fixed, but unknown, distribution D over points (sets) in $2^{[n]}$. The points are labeled by a fixed, but unknown, target function $f^* : 2^{[n]} \rightarrow \mathbb{R}_+$. The goal is to output in polynomial time, with high probability, a hypothesis function f that is a good multiplicative approximation for f^* over most sets with respect to D. Formally, we want to achieve

$$\Pr_{S_1,\ldots,S_m \sim D}\left[\ \Pr_{S \sim D}\left[\, f(S) \leq f^*(S) \leq \alpha f(S) \,\right]\ \geq\ 1-\epsilon\ \right]\ \geq\ 1-\delta$$

for an algorithm that uses $m = \mathrm{poly}(n, \frac{1}{\varepsilon}, \frac{1}{\delta})$ samples and that runs in $\mathrm{poly}(n, \frac{1}{\varepsilon}, \frac{1}{\delta})$ time. That is, in this model one must *approximate* the value of a function on a set of large measure, with high confidence. Asking for low multiplicative error on most points composes naturally with approximation algorithms guarantees. Note that an alternative approach for dealing with real-valued functions is to consider other loss functions such as the squared-loss or the L_1-loss. However, these do not distinguish between the case of having low error on most of the distribution and high error on just a few points, versus moderately high error everywhere. In comparison, the PMAC model allows for more fine-grained control with separate parameters for the amount and extent of errors, and in addition it allows for consideration of multiplicative error which is often more natural in this context.

Our first main result is a (nearly) tight bound of $\alpha = \tilde{\Theta}(\sqrt{n})$ on the learnability of both subadditive and XOS–also known as *fractionally subadditive*—valuations. The class of XOS valuations is a highly expressive subclass of subadditive functions and it has an appealing syntactic description: it is the class of those functions that can be represented as a depth-two tree with a MAX root and SUM inner nodes, where each such SUM node has a subset of items with associated positive weights as leaves. For example, suppose items represent different attractions that a vacation destination might have, such as skiing, hiking, swimming, and sightseeing. A traveler considering different possible travel dates might value this destination as the maximum (over the possible dates) of different linear functions over attractions for those dates (e.g., a summer date having higher value on swimming and a winter date having higher value on skiing). In this paper, we show a nearly tight $O(\sqrt{n})$ upper bound and $\Omega(\sqrt{n}/\log n)$ lower bound on the learnability of XOS valuations in the PMAC model. The key element in proving our upper bound is to show that any XOS function can be approximated by the square root of a linear function within a factor $O(\sqrt{n})$. Using this, we then reduce the problem of PMAC-learning XOS valuations to the standard problem of learning linear separators in the PAC model which can be done efficiently. Our $\Omega(\sqrt{n}/\log n)$ lower bound is information theoretic, applying to *any* procedure that uses a polynomial number of samples. Using the fact that XOS valuations can approximate any valuation in the subadditive superclass up to a $O(\log n)$ factor (Bhawalkar and Roughgarden, 2011; Dobzinski, 2007) and properties of the algorithm we develop for XOS functions, we also show a $O(\sqrt{n}\log n)$ upper bound on the learnability of subadditive valuations. This upper bound improves on the computationally efficient algorithm of Balcan and Harvey (2011) which achieves a worse approximation factor of $O(n)$. It also improves over the concurrent and independent work of Badanidiyuru et al. (2012) which achieves a slightly worse approximation factor (still $\tilde{O}(\sqrt{n})$) but via a computationally inefficient algorithm.

Our second main result is a target-dependent learnability result for XOS functions. Exploiting the fact that XOS functions have a syntactic characterization, we consider learnability as a function of the complexity of the target function in the MAX of SUMs representation and provide efficient algorithms with stronger guarantees for XOS functions of polynomial description length (such as the XOS function in the traveler example above). Specifically, we show the class of XOS functions representable with at most R trees can be PMAC-learned to an $O(R^{\xi})$ factor in time $n^{O(1/\xi)}$ for any $\xi > 0$. In particular, for R polynomial in n, we get learnability to an $O(n^{\xi})$ factor in polynomial time for any constant $\xi > 0$. Technically, we prove this result via a novel structural result showing that a XOS function can be approximated well by the L-th root of a degree-L polynomial over the natural feature representation of the set S, for $L = 1/\xi$. Conceptually, this result highlights the importance of the complexity of the target function for polynomial time learning.[2]

We also consider another class of valuations commonly used in (algorithmic) game theory, namely OXS functions (representable as the SUM of MAX of item values), which include linear valuations. By exploiting novel structural results on approximability with simple functions, we provide much better upper bounds for other interesting subclasses of OXS and XOS. These include OXS and XOS functions with a small number of leaves per tree and OXS functions with a small number of trees. Some of these classes have been considered in the context of economic optimization problems (Babaioff et al., 2007; Bhawalkar and Roughgarden, 2011; Buchfuhrer et al., 2010a), but we are the first to study their learnability.

We show that the structural results we derive for analyzing learnability in the distributional learning setting also have implications for the model of approximate learning everywhere with value queries, analyzed in recent years for several classes of set functions learning by (Blum et al., 2003; Goemans et al., 2009; Svitkina and Fleischer, 2008). In particular, our structural results lead to new upper bounds for XOS and OXS as well as new lower bounds for OXS and gross substitutes.

Finally, we introduce a new model for learning with prices in which the learner receives less information on the values $f^*(S_1), f^*(S_2), \ldots$: for each l, the learner can only *quote* a price p_l and observe whether the agent buys S_l or not, i.e. whether $p_l \leq f^*(S_l)$ or not. This model is more realistic in economic settings where agents interact with a seller via prices only. Interestingly, many of our upper bounds, both for PMAC-learning and learning with value queries, are preserved in this model (all lower bounds automatically continue to hold).

Related Work on Learning Valuations Balcan and Harvey (2011) analyze the PMAC learnability of submodular functions (a subclass of XOS functions and superclass of OXS functions) and subadditive functions. They provide a $O(\sqrt{n})$ upper bound for submodular functions and $O(n)$ upper bound for subadditive functions. We improve upon the latter result by providing a $\tilde{O}(n^{1/2})$ upper bound for subadditive functions, matching their result for the less general class of submodular functions. Balcan and Harvey (2011) also show a technically involved lower bound of $\Omega(n^{1/3}/\log n)$ for matroid rank functions (and thus submodular functions). We exploit the power of XOS functions and provide a much simpler $\Omega(n^{1/2}/\log n)$ lower bound for such functions.

In concurrent and independent work, Badanidiyuru et al. (2012) provide several interesting (related, but largely complementary) results concerning both sketching and PMAC learning of several subclasses of subadditive valuations.They focus on sketching valuation functions, where the goal is to produce a hypothesis that can be represented in $\mathrm{poly}(n)$ space and that approximates the target function at every point to within a multiplicative factor; that is, the goal is identical with that of our model of approximate learning everywhere with value queries, but the queries allowed are more

[2] Note that since the class of XOS functions representable with at most a polynomial number of trees has small complexity, learnability would be immediate if we did not care about computational efficiency.

general. Badanidiyuru et al. (2012) provide an algorithm that uses a polynomial number of demand queries [3] and outputs a sketch with an $\alpha = \tilde{O}(\sqrt{n})$ factor for XOS and subadditive functions. This is a stronger (*i.e.*, pointwise) guarantee than our PMAC learning result for these classes, but it uses demand queries which are more powerful than value queries. They also show that, ignoring computational considerations, sketching implies PMAC learning, and as a consequence, they obtain an information theoretic upper bound of $\tilde{O}(\sqrt{n})$ factor for PMAC-learning of XOS and subadditive functions. In comparison, our PMAC learning result for these classes is computationally efficient and moreover, it achieves a slightly better bound. Badanidiyuru et al. (2012) also provide several lower bounds, some of them analogous to the ones we derive in this paper.

A few recent papers (Goemans et al., 2009; Svitkina and Fleischer, 2008) consider learnability within the approximate learning everywhere with value queries model, but for submodular functions only. Other models for the value queries paradigm require exact learning and are necessarily limited to much less general function classes than the ones we study here: read-once and Toolbox DNF valuations (Blum et al., 2003), polynomial or linear-threshold valuations (Lahaie and Parkes, 2004) or MAX or SUM (of bundles) valuations (Lahaie et al., 2005).

Finally, recent work of Vainsencher et al. (2011) considers a quite different and online selection problem of bundles and shows that under the restriction that only pair-wise interaction affect the valuation, that valuations can be elicited in a difficult online scenario. The types of valuation functions we consider in this paper (though technically incomparable) are more general.

Other Applications Beyond learning valuation functions, the problem of learning set functions of the type we consider in this paper is natural for a wide range of settings where, based on historical data, one would like to predict the value of some function over objects described by features. Examples include predicting the rate of growth of jobs (or population) in cities as a function of various amenities or enticements that the city offers, predicting the sales price of a house as a function of features (such as an updated kitchen, hardwood floors, extra bedrooms, etc.) that it might have, or predicting the spread of infectious diseases in animals as a function of various features of the population, landscape, and predators. In such settings it is natural to assume various degrees of no-complementarities (OXS, XOS, subadditive), making them well-suited to a formulation as a problem of learning set functions of the type we study.

Discussion We note that all our upper bounds are efficient and all the lower bounds are information theoretic. We remark that it is quite interesting one can get a nearly tight, sublinear approximation efficiently (that is, in time polynomial in n only) for general XOS and subadditive functions, since these functions can

Due to lack of space, we only include proof sketches in the main body, with further details in the appendices. Our results for approximate learning everywhere with value queries appear in Appendix C and our results for the learning with prices model appear in Appendix E.

2. Setup

We consider a universe $[n] = \{1, \ldots, n\}$ of items and *valuations*, i.e. monotone non-negative set functions $f : 2^{[n]} \to \mathbb{R}_+$: $f(S \cup \{i\}) \geq f(S) \geq 0, \forall S \subseteq [n], \forall i \notin S$. For a set $S \subseteq [n]$ we denote

[3] In a demand query, an agent is presented with prices for each item, and is required to report a subset of items that maximizes his utility. For certain classes such as OXS functions, a demand query can be simulated by a polynomial number of value queries, but in general they are strictly more powerful than value queries.

by $\chi(S) \in \{0,1\}^n$ its indicator vector; so $(\chi(S))_i = 1$ if $i \in S$ and $(\chi(S))_i = 0$ if $i \notin S$. We often use this natural isomorphism between $\{0,1\}^n$ and $2^{[n]}$.

Learning Model The main model we consider in this paper is a passive supervised learning model of real valued functions with a multiplicative approximation (Balcan and Harvey, 2011). We assume that the input for a 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 $2^{[n]}$. The points are labeled by a fixed, but unknown, target function $f^* : 2^{[n]} \to \mathbb{R}_+$. The goal is to output a hypothesis function f such that, with large probability over the choice of examples, the set of points for which f is a good approximation for f^* has large measure with respect to D. Formally:

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

$$\Pr_{S_1,\dots,S_m\sim D}\left[\Pr_{S\sim D}\left[\, f(S) \leq f^*(S) \leq \alpha f(S)\,\right] \geq 1-\varepsilon\,\right] \geq 1-\delta \tag{1}$$

The number m of samples used by $\mathcal{A}$ is poly$(n, \frac{1}{\varepsilon}, \frac{1}{\delta})$. If $\mathcal{A}$'s running time is poly$(n, \frac{1}{\varepsilon}, \frac{1}{\delta})$, then we say that $\mathcal{A}$ efficiently learns $\mathcal{F}$.

PMAC stands for Probably Mostly Approximately Correct. In this model, one must *approximate* the value of a function on a set of large measure, with high confidence. Note that the traditional PAC model requires one to predict the value *exactly* on a set of large measure, with high confidence. The PAC model (Valiant, 1984) is the special case of this model with $\alpha = 1$.

In Appendix C we also provide results on approximate learning everywhere with value queries.

Classes of Valuation Functions. We now define and give intuition for most valuation classes we focus on. The most general type of valuations we consider, subadditive valuations model the lack of synergies among sets: a set's value is at most the sum of the values of its parts. Formally:

Definition 2 *$f : 2^{[n]} \to \mathbb{R}_+$ is* subadditive *if and only if $f(S \cup S') \leq f(S) + f(S'), \forall S, S' \subseteq [n]$.*

XOS is an important class of subadditive valuations studied in (algorithmic) game theory and economics (Dobzinski et al., 2005, 2006; Feige, 2006; Lehmann et al., 2001). A valuation is XOS iff it can be represented as a depth-two tree with a MAX root and SUM inner nodes. Each such SUM node has as leaves a subset of items with associated positive weights. For example, a traveler may choose the destination of maximum value among several different locations, where each location has a number of amenities and the valuation for a location is linear in the set of amenities.

Definition 3 *$f : 2^{[n]} \to \mathbb{R}_+$ is* XOS *if and only if it can be represented as the maximum of k linear valuations, for some $k \geq 1$. That is, $f(S) = \max_{j=1\dots k} w_j^T \chi(S)$ where $w_{ji} \geq 0, \forall j \in [k], \forall i \in [n]$.*

We say that item i appears as a leaf in a SUM tree j if i has a positive value in tree j.

When reversing the roles of operators MAX and SUM we obtain a strict *sub*class of XOS valuations, called OXS,[4] that is also relevant to economics (Day and Raghavan, 2006; Dobzinski

[4] XOS and OXS stand for XOR-of-OR-of-Singletons and OR-of-XOR-of-Singletons, where MAX is denoted by XOR and SUM by OR(Nisan, 2006; Sandholm, 2001).

et al., 2006; Lehmann et al., 2001; Singer, 2010). To define OXS we also define a unit-demand valuation, in which the value of any set S is the highest weight of any item in S. A unit-demand valuation is essentially a tree, with a MAX root and one leaf for each item with non-zero associated weight. In an OXS valuation, a set's value is given by the best way to split the set among several unit-demand valuations. An OXS valuation f has a natural representation as a depth-two tree, with a SUM node at the root (on level 0), and subtrees[5] corresponding to the unit-demand valuations $f_1, \dots, f_k$. The value $f(S)$ of any set S corresponds to best way of partitioning S into $(S_1, \dots, S_k)$ and adding up the per-tree values $\{f_1(S_1), \dots, f_k(S_k)\}$.

Definition 4 *A* unit-demand valuation f *is given by weights* $\{w_1, ..., w_n\} \subset \mathbb{R}_+$ *such that* $f(S) = \max_{i \in S} w_i, \forall S \subseteq [n]$. *An* OXS valuation f *is given by the convolution of* $k \geq 1$ *unit-demand valuations* $f_1, \dots, f_k$*: that is,*
$f(S) = \max\{f_1(S_1) + \cdots + f_k(S_k) : (S_1, \dots, S_k)$ *is a partition of* $S\}, \forall S \subseteq [n]$.

Two other common classes of valuation functions include submodular valuations and gross substitutes (GS). The classes of valuations discussed so far form a strict hierarchy. (See (Lehmann et al., 2001) for examples separating these classes of valuations.)

Lemma 1 *(Lehmann et al., 2001)* OXS $\subsetneq$ *gross substitutes* $\subsetneq$ *submodular* $\subsetneq$ XOS $\subsetneq$ *subadditive.*

3. PMAC-learnability of XOS valuations and subadditive valuations

In this section we give nearly tight lower and upper bounds of $\tilde{\Theta}(\sqrt{n})$ for the PMAC-learnability of XOS and subadditive valuations, as well as improved learnability guarantees for interesting special classes of XOS functions.

3.1. Nearly tight lower and upper bounds for learning XOS and subadditive functions

We establish our $\tilde{\Theta}(\sqrt{n})$ bounds by showing an $\Omega(\sqrt{n}/\log n)$ lower bound for the class of XOS valuations (hence valid for subadditive valuations) and upper bounds of $O(\sqrt{n})$ and $O(\sqrt{n}\log n)$ for the classes of XOS and subadditive valuations respectively.

Theorem 1 *The classes of* XOS *and subadditive functions are PMAC-learnable to a* $\tilde{\Theta}(\sqrt{n})$ *approximation factor.*

Proof Sketch: **Lower bound**: We start with an information theoretic lower bound showing that the class of XOS valuations cannot be learned with an approximation factor of $o(\frac{\sqrt{n}}{\log n})$ from a polynomial number of samples. Let $k = n^{\frac{1}{3}\log\log n}$. For large enough n we can show that there exists a family of sets $\mathcal{A} = \{A_1, \dots, A_k\}$, $A_1, A_2, ..., A_k \subseteq [n]$ such that we have both
(i) $\sqrt{n}/2 \leq |A_i| \leq 2\sqrt{n}$ for any $1 \leq i \leq k$, i.e. all sets have large size $\Theta(\sqrt{n})$ and
(ii) $|A_i \cap A_j| \leq \log n$ for any $1 \leq i < j \leq k$, i.e. all pairwise intersections have small size $O(\log n)$.

We achieve this via a simple probabilistic argument, constructing each A_i by picking each element in $[n]$ with probability $\frac{1}{\sqrt{n}}$ — see Lemma 1 in Appendix A. Given the existence of family $\mathcal{A}$

[5] Another OXS encoding uses a weighted bipartite graph $G_{n,k}$ where edge (i, j) has the weight of item i in f_j; $f(S)$ is the weight of a maximum matching of S to the k nodes for the unit-demand f_j's. Also, OXS valuations with weights $\{0,1\}$ are exactly rank functions of transversal matroids.

we construct a hard family of XOS functions as follows. For any subfamily $\mathcal{B} \subseteq \mathcal{A}$, we construct an XOS function $f_{\mathcal{B}}$ with large values for sets $A_i \in \mathcal{B}$ and small values for sets $A_i \notin \mathcal{B}$. Let $h_{A_i}(S) = |S \cap A_i|$ for any $S \subseteq [n]$. For any subfamily $\mathcal{B} \subseteq \mathcal{A}$, define the XOS function $f_{\mathcal{B}}$ by $f_{\mathcal{B}}(S) = \text{MAX}_{\ A_i \in \mathcal{B}}\, h_{A_i}(S)$. We claim that $f_{\mathcal{B}}(A_i) = \Omega(\sqrt{n})$, if $A_i \in \mathcal{B}$ but $f_{\mathcal{B}}(A_i) = O(\log n)$, if $A_i \notin \mathcal{B}$. Indeed, for any $A_i \in \mathcal{B}$, we have $h_{A_i}(A_i) = |A_i| \geq \sqrt{n}/2$, hence $f_{\mathcal{B}}(A_i) = \Omega(\sqrt{n})$; for any $A_j \notin \mathcal{B}$, by our construction of $\mathcal{A}$, we have $h_{A_i}(A_j) = |A_i \cap A_j| \leq \log n$, implying $f_{\mathcal{B}}(A_j) = O(\log n)$. For an unknown $\mathcal{B}$, the problem of learning $f_{\mathcal{B}}$ within a factor of $o(\sqrt{n}/\log n)$ under a uniform distribution on $\mathcal{A}$ amounts to distinguishing $\mathcal{B}$ from $\mathcal{A}$. This is not possible from a polynomial number of samples since $|\mathcal{A}| = n^{\frac{1}{3}\log\log n}$. In particular, if $\mathcal{B} \subseteq \mathcal{A}$ is chosen at random, then any algorithm from a polynomial-sized sample will have error $\Omega(\frac{\sqrt{n}}{\log n})$ on a region of probability mass greater than $\frac{1}{2} - \frac{1}{\text{poly}(n)}$.

Upper bounds: We show that the class of XOS valuations can be PMAC-learned to a $O(\sqrt{n})$ factor and that the class of subadditive valuations can be PMAC-learned to a $O(\sqrt{n}\log n)$ factor, by using $O(\frac{n}{\epsilon}\log\frac{n}{\delta\epsilon})$ training examples and running time $\text{poly}(n, \frac{1}{\varepsilon}, \frac{1}{\delta})$. To prove these bounds we start by providing a structural result (Claim 1 below) showing that XOS valuations can be approximated to a $\sqrt{n}$ factor by the square root of a linear function.

Claim 1 *Let $f : 2^{[n]} \to \mathbb{R}_+$ be a non-negative* XOS *function with $f(\emptyset) = 0$. Then there exists $\hat{f}$ of the form $\hat{f}(S) = \sqrt{w^{\mathsf{T}}\chi(S)}$ where $w \in \mathbb{R}^n_+$ such that $\hat{f}(S) \leq f(S) \leq \sqrt{n}\hat{f}(S)$ for all $S \subseteq [n]$.*

Proof First, we note that XOS valuations are known to be equivalent to fractionally subadditive valuations (Feige, 2006). A function $f : 2^{[n]} \to \mathbb{R}$ is called *fractionally subadditive* if $f(T) \leq \sum_S \lambda_S f(S)$ whenever $\lambda_S \geq 0$ and $\sum_{S: s \in S} \lambda_S \geq 1$ for any $s \in T$. Second, we exploit the fact that a fractionally subadditive valuation f satisfies $f(T) = \max\{\sum_{i\in T} x_i | x \in P(f)\}$ for any $T \subseteq [n]$, where $P(f)$ is the associated polyhedron $\{x \in \mathbf{R}^n_+ : \sum_{i \in S} x_i \leq f(S), \forall S \subseteq [n]\}$. Informally, this result states that one recovers $f(T)$ when optimizing in the direction given by T over the polyhedron $P(f)$ associated with f. The proof of this result involves a pair of dual linear programs, one corresponding to the maximization and another one that is tailored for fractional subadditivity, with an optimal objective value of $f(T)$ (Feige, 2006). For completeness, we include the formal proof of this result in Appendix A (see Lemma 2). Given this result, we proceed as follows (a similar approach has been used by Goemans et al. (2009) for submodular functions.). Define $P = \{x \in \mathbf{R}^n : (|x_1|, ..., |x_n|) \in P(f)\}$. Since P is bounded and central symmetric (i.e. $x \in P \Leftrightarrow -x \in P$), there exists (John, 1948) an ellipsoid $\mathcal{E}$ containing P such that $\frac{1}{\sqrt{n}}\mathcal{E}$ is contained in P. Hence for $\hat{f}(T) = \max\{\sum_{i\in T} x_i : x \in \frac{1}{\sqrt{n}}\mathcal{E}\}$, we have $\hat{f}(T) \leq f(T) \leq \sqrt{n}\hat{f}(T), \forall T \subseteq [n]$. Moreover, the ellipsoid $\mathcal{E}$ is axis-alligned, and basic calculus implies $\hat{f}(T) = \sqrt{w^{\mathsf{T}}\chi(T)}$ for some $w \in \mathbb{R}^n_+$. ■

For PMAC-learning XOS valuations to with an approximation factor of $\sqrt{n+\varepsilon}$, we apply Algorithm 1 with parameters $R = n$, ϵ, and $p = 2$. The proof of correctness of Algorithm 1 follows by using the structural result in Claim 1 and a technique of Balcan and Harvey (2011) that we sketch briefly here. Full details of this proof appear in Appendix A.1.

Assume first that $f^*(S) > 0$ for all $S \neq \emptyset$. The key idea is that Claim 1's structural result implies that the following examples in $\mathbb{R}^{n+1}$ are linearly separable since $nw^{\mathsf{T}}\chi(S) - (f^*(S))^2 \geq 0$

and $nw^{\mathsf{T}}\chi(S) - (n+\epsilon)(f^*(S))^2 < 0$.

$$\begin{array}{lll} \text{Examples labeled } +1: & \mathrm{ex}_S^+ := (\chi(S), (f^*(S))^2) & \forall S \subseteq [n] \\ \text{Examples labeled } -1: & \mathrm{ex}_S^- := (\chi(S), (n+\epsilon)\cdot(f^*(S))^2) & \forall S \subseteq [n] \end{array}$$

This suggests trying to reduce our learning problem to the standard problem of learning a linear separator for these examples in the standard PAC model (Kearns and Vazirani, 1994; Vapnik, 1998). However, in order to apply standard techniques to learn such a linear separator, we must ensure that our training examples are i.i.d. To achieve this, we create a i.i.d. distribution D' in $\mathbb{R}^{n+1}$ that is related to the original distribution D as follows. First, we draw a sample $S \subseteq [n]$ from the distribution D and then flip a fair coin for each. The sample from D' is labeled ex_S^+ i.e. $+1$ if the coin is heads and ex_S^- i.e. -1 if the coin is tails. As mentioned above, these labeled examples are linearly separable in $\mathbb{R}^{n+1}$. Conversely, suppose we can find a linear separator that classifies most of the examples coming from D' correctly. Assume that this linear separator in $\mathbb{R}^{n+1}$ is defined by the function $u^{\mathsf{T}}x = 0$, where $u = (\hat{w}, -z)$, $w \in \mathbb{R}^n$ and $z > 0$. The key observation is that the function $f(S) = \frac{1}{(n+\epsilon)z}\hat{w}^{\mathsf{T}}\chi(S)$ approximates $(f^*(\cdot))^2$ to within a factor $n+\epsilon$ on most of the points coming from D.

If f^* is zero on non-empty sets, then we can learn its set $\mathcal{Z} = \{\, S : f^*(S) = 0 \,\}$ of zeros quickly since $\mathcal{Z}$ is closed to union and taking subsets for any subadditive f^*. In particular, suppose that there is at least an ϵ chance that a new example is a zero of f^*, but does not lie in the null subcube over the sample. Then such a example should be seen in the next sequence of $\log(1/\delta)/\epsilon$ examples, with probability at least $1-\delta$. This new example increases the dimension of the null subcube by at least one, and therefore this can happen at most n times.

To establish learnability for the class of subadditive valuations, we use the fact that any subadditive valuation can be approximated by an XOS valuation to a $\ln n$ factor (Dobzinski, 2007; Bhawalkar and Roughgarden, 2011)[6] and so, by Claim 1, any subadditive valuation is approximated to a $\sqrt{n}\ln n$ factor by a linear function. This then implies that we can use Algorithm 1 with parameters $R = n\ln^2 n$, ϵ, and $p = 2$. Correctness then follows by a reasoning similar to the one for XOS functions. ■

Note: Balcan and Harvey (2011) showed a much more technically involved and slightly weaker lower bound of $\Omega(n^{1/3}/\log n)$, but for the much more restricted class of matroid rank functions. As matroid rank functions are known to be submodular and even satisfy the gross substitutes property (Murota, 2003), their lower bound immediately applies to these classes as well. By contrast, we use the power of XOS functions and provide a much simpler lower bound for such functions.

3.2. Better learnability results for XOS valuations with polynomial complexity

In this section we consider the learnability of XOS valuations representable with a polynomial number of trees. Such functions arise often in practical applications[7] and interestingly we can show that we can achieve good PMAC learnability via polynomial time algorithms. (Since this class has small complexity, it is easy to see that it is learnable in principle from a small sample size if we did

[6] We are grateful to Shahar Dobzinski and Kshipra Bhawalkar for pointing out this fact to us.

[7] Consider a buyer that owns a computer and that is interested in choosing an operating system together with software applications for it. The buyer can choose between several operating systems (e.g. Unix, Macintosh or Windows) and for each system there may a large number of applications. The valuation for a set of applications is the sum of all values for individual applications. This valuation can be represented as an XOS function where each of the several OR trees stands for an operating system.

```
Algorithm 1 Algorithm for PMAC-learning via a reduction to a binary linear separator problem.
Input: Parameters: $R$, $\epsilon$ and $p$. Training examples $\mathcal{S} = \{(S_1, f^*(S_1)), \dots, (S_m, f^*(S_m))\}$.

  • Let $\mathcal{S}_{\neq 0} = \{(A_i, f^*(A_i)) \in \mathcal{S} \colon f^*(A_i) \neq 0\} \subseteq \mathcal{S}$ the examples with non-zero values, $\mathcal{S}_0 = \mathcal{S} \setminus \mathcal{S}_{\neq 0}$ and $\mathcal{U}_0 = \cup_{l \leq m; f^*(S_l)=0} S_l$.
  • For each $i$ in $\{1, \dots, |\mathcal{S}_{\neq 0}|\}$ let $y_i$ be the outcome of independently flipping a fair $\{+1, -1\}$-valued coin.
      Let $x_i \in \mathbb{R}^{n+1}$ be the point defined by $x_i = \begin{cases} (\ \chi(A_i), (f^*(A_i))^p\ ) & \text{(if } y_i = +1\text{)} \\ (\ \chi(A_i), (R+\epsilon) \cdot (f^*(A_i))^p\ ) & \text{(if } y_i = -1\text{).} \end{cases}$
  • Find a linear separator $u = (\hat{w}, -z) \in \mathbb{R}^{n+1}$, where $\hat{w} \in \mathbb{R}^n$ and $z > 0$, such that $(x, \mathrm{sgn}(u^\mathsf{T} x))$ is consistent with the labeled examples $(x_i, y_i)\ \forall i \in \{1, \dots, |\mathcal{S}_{\neq 0}|\}$, and with the additional constraint that $\hat{w}_j = 0\ \forall j \in \mathcal{U}_0$.

Output: The function $f$ defined as $f(S) = \left( \frac{1}{(R+\epsilon)z}\, \hat{w}^\mathsf{T} \chi(S) \right)^{1/p}$.
```

not care about computational complexity.) In particular, we show that XOS functions representable with at most R SUM trees can be PMAC-learned with a R^{ξ} approximation factor in time $n^{O(1/\xi)}$, for any $\xi > 0$. This improves the approximation factor of Theorem 1 for all such XOS functions. Moreover, this implies that XOS valuations representable with a polynomial number of trees can be PMAC-learned within a factor of n^{ξ}, in time $n^{O(1/\xi)}$, for any $\xi > 0$.

Theorem 2 *For any $\xi > 0$, the class of* XOS *functions representable with at most $R = n^{O(1)}$* SUM *trees is PMAC-learnable in time $n^{O(1/\xi)}$ with approximation factor of $(R + \varepsilon)^{\xi}$ by using $O\left(\frac{n^{1/\xi}}{\epsilon} \left[\frac{\log(n)}{\xi} + \log\left(\frac{1}{\delta\epsilon}\right) \right] \right)$ training examples.*

Proof Let $L = 1/\xi$ and assume for simplicity that it is integer. We start by deriving a key structural result. We show that XOS functions can be approximated well by the L-th root of a degree-L polynomial over $(\chi(S))_i$ for $i \in [n]$. Let $T_1, \dots, T_R$ be the R SUM trees in an XOS representation $\mathcal{T}$ of f^*. For a tree j and a leaf in T_j corresponding to an element $i \in [n]$, let w_{ji} the weight of the leaf. For any set S, let $k_j(S) = \sum_{i \in T_j \cap S} w_{ji} = w_j^T \chi(S)$ be the sum of weights in tree T_j corresponding to leaves in S. $k_j(S)$ is the value assigned to set S by tree T_j. Note that $f^*(S) = \max_j k_j(S)$, i.e. the maximum value of any tree, from the definition of MAX . We define valuation f' that averages the L-th powers of the values of all trees: $f'(S) = 1/R \sum_j k_j^L(S)$, $\forall S \subseteq [n]$. We claim that $f'(\cdot)$ approximates $(f^*(\cdot))^L$ to within an R factor on all sets S, namely

$$f'(S) \leq (f^*(S))^L \leq R f'(S),\ \forall S \subseteq [n] \quad \text{i.e.} \tag{2}$$

$$1/R \textstyle\sum_j k_j^L(S) \leq \max_j k_j^L(S) \leq \textstyle\sum_j k_j^L(S),\ \forall S \subseteq [n] \tag{3}$$

The left-hand side inequalities in Eq. (3) follow as f^* has at most R trees and $k_{j'}^L(S) \leq \max_j k_j^L(S)$ for any tree $T_{j'}$. The right-hand side inequalities in Eq. (3) follow immediately.

This structural result suggests re-representing each set S by a new set of $\Theta(n^L)$ features, with one feature for each subset of $[n]$ with at most L items. Formally, for any set $S \subseteq [n]$, we denote by $\chi_M(S)$ its feature representation over this new set of features. $\chi_M(S)_{i_1, i_2, \dots, i_L} = 1$ if all items

$i_1, i_2, \ldots i_L$ appear in S and $\chi_M(S)_{i_1,i_2,\ldots,i_L} = 0$ otherwise. It is easy to see that f' is representable as a linear function over this new set of features. This holds for each $k_j^L(S) = (w_j^T \chi(S))^L$ due to its multinomial expansion, that contains one term for each set of up to L items appearing in tree T_j, i.e. for each such feature. Furthermore, f' remains linear when the terms for each tree T_j are added.

Given this, we can now use a variant of Algorithm 1 with parameters R, ϵ, and $p = L$ and to prove correctness we can use a reasoning similar to the one in Theorem 1. Any sample S_l is fed into Algorithm 1 as $(\chi_M(S_l), (f^*(S_l))^L)$ or $(\chi_M(S_l), (R+\epsilon)\cdot(f^*(S_l))^L)$ respectively. Since f' is linear over the set of features, Algorithm 1 outputs with probability at least $1-\delta$ a hypothesis f'' that approximates f^* to an $(R+\varepsilon)^{1/L}$ factor on any point $\chi_M(S)$ corresponding to sets $S \subseteq [n]$ from a collection $\mathcal{S}$ with at least an $1-\varepsilon$ measure in D, i.e. $f''(\chi_M(S)) \leq f^*(S) \leq (R+\varepsilon)^{1/L} f''(\chi_M(S))$. We can output then hypothesis $f(S) = f''(\chi_M(S)), \forall S \subseteq [n]$, defined on the initial ground set $[n]$ of items, that approximates $f^*(\cdot)$ well, i.e., as desired, for any $S \in \mathcal{S}$ we have

$$f(S) = f''(\chi_M(S)) \leq f^*(S) \leq (R+\varepsilon)^{1/L} f''(\chi_M(S)) = (R+\varepsilon)^{1/L} f(S).$$

As desired, with high confidence the hypothesis f approximates f^* to a $(R+\varepsilon)^{\xi}$ factor on most sets from D. ■

Note: This result also has an appealing interpretation for submodular functions. It is known that any submodular function is representable as an XOS tree (Lehmann et al., 2001), though with possibly exponential number of SUM trees. What Theorem 2 implies that (submodular) functions that are succinctly representable as XOS trees can be PMAC-learned well.

3.3. Better learnability results for XOS valuations with small SUM trees

In this section we consider the learnability of another interesting subclass of XOS valuations, namely XOS valuations representable with "small" SUM trees and show learnability to a better factor than that in Theorem 1. [8]

Theorem 3 *For any $\eta > 0$, the class of* XOS *functions representable with* SUM *trees with at most R leaves is properly PMAC-learnable with approximation factor of $R(1+\eta)$ by using $m = O(\frac{1}{\epsilon}\left(n \log\log_{1+\eta}(\frac{H}{h}) + \log(1/\delta)\right))$ and running time polynomial in m, where h and H are the smallest and the largest non-zero values our functions can take.*

Proof Algorithmically, we construct a (unit-demand) hypothesis function f as follows. For any i that appears in at least one set S_j in the sample we define $f(i)$ as the smallest value $f^*(S_j)$ over all the sets S_j in the sample containing i. For i that does not appear in any set S_j define $f(i) = 0$. We then define $f(S) = \max_{i \in S} f(i)$ for any $S \subseteq \{1, \ldots, n\}$. See Algorithm 2 for a formal description.

We start by proving a key structural result showing that f approximates the target function multiplicatively within a factor of R over the sample. That means:

$$f(S_l) \leq f^*(S_l) \leq R f(S_l) \quad \text{for all } \; l \in \{1, 2, \ldots, m\}. \tag{4}$$

[8] For example, consider a traveler deciding between many trips, each to a different location with a small number of tourist attractions. The traveler has an additive value for several attractions at the same location. This valuation can be represented as an XOS function where each SUM tree stands for a location and has a small number of leaves.

Algorithm 2 Algorithm for PMAC-learning interesting classes of XOS and OXS valuations.

```
Input: A sequence of training examples $\mathcal{S} = \{(S_1, f^*(S_1)), (S_2, f^*(S_2)), \ldots (S_m, f^*(S_m))\}$.
    • Set $f(i) = \min_{j: i \in S_j} f^*(S_j)$ if $i \in \cup_{l=1}^{m} S_l$ and $f(i) = 0$ if $i \notin \cup_{l=1}^{m} S_l$.
Output: The unit-demand valuation $f$ defined by $f(S) = \max_{i \in S} f(i)$ for any $S \subseteq \{1, \ldots, n\}$.
```

To see this note that for any $i \in S_l$ we have $f^*(i) \leq f^*(S_l)$, for $l \in \{1, 2, \ldots, m\}$. So

$$f(i) \geq f^*(i) \quad \text{for any } i \in S_1 \cup \ldots \cup S_m. \tag{5}$$

Therefore for any $l \in \{1, 2, \ldots, m\}$. : $f^*(S_l) \leq R \max_{i \in S_l} f^*(i) \leq R \max_{i \in S_l} f(i) = Rf(S_l)$, where the first inequality follows by definition, and the second inequality follow from relation (5). By definition, for any $i \in S_l$, $f(i) \leq f^*(S_l)$. Thus, $f(S_l) = \max_{i \in S_l} f(i) \leq f^*(S_l)$. These together imply relation (4), as desired.

To finish the proof we show that $m = O(\frac{1}{\epsilon}\left(n \log \log_{1+\eta}(\frac{H}{h}) + \log(1/\delta)\right))$ is sufficient so that with probability $\geq 1 - \delta$ f approximates the target function f^* multiplicatively within a factor of $R(1+\eta)^2$ on a $1 - \epsilon$ fraction of the distribution. Let F_η be the class of unit-demand functions that assign to each individual leaf a power of $(1+\eta)$ in $[h, H]$. Clearly $|F_\eta| = (\log_{1+\eta}(\frac{H}{h}))^n$. It is easy to see that $m = O(\frac{1}{\epsilon}\left(n \log \log_{1+\eta}(\frac{H}{h}) + \log(1/\delta)\right))$ examples are sufficient s.t. any function in F_η that approximates the target function on the sample multiplicatively within a factor of $R(1+\eta)$ will with probability at least $1 - \delta$ approximate the target function multiplicatively within a factor of $R(1+\eta)$ on a $1 - \epsilon$ fraction of the distribution. Since F_η is a multiplicative L_∞ cover for the class of unit-demand functions, we easily get the desired result (Anthony and Bartlett, 1999). ■

4. PMAC-learnability of OXS valuations

In this section we study the learnability of OXS valuations, focusing on interesting subclasses of OXS functions that arise in practice, namely OXS functions representable with a small number of MAX trees or leaves[9]– see Appendix B for motivating examples.

Theorem 4 *(1) Let $\mathcal{F}$ be the family of* OXS *functions representable with at most R* MAX *trees. For any η, the family $\mathcal{F}$ is properly PMAC-learnable with approximation factor of $R(1+\eta)$ by using $m = O(\frac{1}{\epsilon}\left(n \log \log_{1+\eta}(\frac{H}{h}) + \log(1/\delta)\right))$ training examples and running time polynomial in m, where h and H are the smallest and the largest value our functions can take. For constant R, the class $\mathcal{F}$ is PAC-learnable by using $O(n^R \log(n/\delta)/\varepsilon)$ training examples and running time* $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$.

(2) For any $\epsilon > 0$, the class of OXS *functions representable with* MAX *trees with at most R leaves is PMAC-learnable with approximation factor $R + \epsilon$ by using $O(\frac{n}{\epsilon} \log\left(\frac{n}{\delta\epsilon}\right))$ training examples and running time* $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$.

Proof [Proof sketch] **(1)** We can show that a function f with an OXS representation $\mathcal{T}$ with at most R trees can also be represented as an XOS function with at most R leaves per tree. Indeed, for

[9] We note that the literature on algorithms for secretary problems (Babaioff et al., 2009, 2007) often considers a subclass of the latter class, in which each item must have the same value in any tree.

each tuple of leaves, one from each tree in $\mathcal{T}$, we create an SUM tree with these leaves. The XOS representation of f^* is the MAX of all these trees. Given this the fact that $\mathcal{F}$ is learnable to a factor of of $R(1+\eta)$ for any η follows from Theorem 3. We now show that when R is constant the class $\mathcal{F}$ is PAC-learnable. First, using a similar argument to the one in Theorem 3 we can show that Algorithm 2 can be used to PAC-learn any unit-demand valuation by using $m = O(n \ln(n/\delta)/\varepsilon)$ training examples and time $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$ – see Lemma 2 in Appendix B. Second, it is easy to see that an OXS function f^* representable with at most R trees can also be represented as a unit-demand with at most n^R leaves, with R-tuples as items (see Lemma 3 in Appendix B). These two facts together imply that for constant R, the class $\mathcal{F}$ is PAC-learnable by using $O(n^R \log(n/\delta)/\varepsilon)$ training examples and running time $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$.

(2) We start by showing the following structural result: if f^* has an OXS representation with at most R leaves in any MAX tree, then it can be approximated by a linear function within a factor of R on every subset of the ground set. In particular, the linear function f defined as $f(S) = \sum_{i \in S} f^*(i)$, for all $S \subseteq \{1 \ldots n\}$ satisfies

$$f^*(S) \leq f(S) \leq R \cdot f^*(S) \quad \text{for all} \quad S \subseteq \{1 \ldots n\} \tag{6}$$

By subadditivity, $f^*(S) \leq Rf(S)$, for all S. Let ${f^*}_1, \ldots {f^*}_k$ be the unit-demand functions that define f^*. Fix a set $S \subseteq [n]$. For any item $i \in S$, define j_i to be the index of the ${f^*}_j$ under which item i has highest value: $f^*(i) = {f^*}_{j_i}(\{i\})$. Then for the partition $(S_1, ..., S_k)$ of S in which item i is mapped to S_{j_i} for any i, we have $\sum_{i \in S} f^*(i) \leq R{f^*}_1(S_1) + \ldots R{f^*}_k(S_k)$. Therefore:

$$f(S) = \tfrac{1}{R} \sum_{i \in S} f^*(i) \ \leq \ \max_{(S_1,\ldots,S_k) \text{ partition of } S} \ ({f^*}_1(S_1) + \cdots + {f^*}_k(S_k)) \ = \ f^*(S),$$

where the last equality follows simply from the definition of an OXS function.

Given the structural result 6, we can PMAC-learn the class of OXS functions representable with MAX trees with at most R leaves y using Algorithm 1 with parameters R, ϵ and $p = 1$. The correctness by using a reasoning similar to the one in Theorem 1. ■

5. Discussion and open questions

Our most general results provide (nearly) tight bounds on the learnability of subadditive and fractionally subadditive valuations. We additionally leverage the structure of valuations in a number of interesting subclasses and obtain algorithms with stronger learning guarantees. However, none of the guarantees for the subclasses are known to be tight. It is an interesting open question to provide tight approximation factors for these subclasses. More generally, our work provides the first target dependent learnability results for several interesting subclasses of subadditive and submodular functions. It would be interesting to further exploit this angle both in the context of learning and combinatorial optimization. In particular, it would be interesting to provide and explore other natural representations of submodular (and more generally subadditive) functions, and provide both learning and optimization procedures with better guarantees for functions that have low complexity under that representation.

Acknowledgments We thank Avrim Blum, Nick Harvey, and David Parkes for useful discussions.

This work was supported in part by NSF grants CCF-0953192 and CCF-1101215, AFOSR grant FA9550-09-1-0538, ONR grant N00014-09-1-0751, and a Microsoft Research Faculty Fellowship.

References

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

M. Babaioff, N. Immorlica, and R. Kleinberg. Matroids, secretary problems, and online mechanisms. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.

M. Babaioff, M. Dinitz, A. Gupta, N. Immorlica, and K. Talwar. Secretary problems: weights and discounts. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2009.

A. Badanidiyuru, S. Dobzinski, H. Fu, R. D. Kleinberg, N. Nisan, and T. Roughgarden. Sketching valuation functions. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2012.

M. F. Balcan and N. Harvey. Learning submodular functions. In *Proceedings of 43rd ACM Symposium on Theory of Computing*, 2011.

K. Bhawalkar and T. Roughgarden. Welfare guarantees for combinatorial auctions with item bidding. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2011.

A. Blum, M. Zinkevich, and T. Sandholm. On polynomial-time preference elicitation with value queries. In *ACM Conference on Electronic Commerce*, 2003.

D. Buchfuhrer, S. Dughmi, H. Fu, R. Kleinberg, E. Mossel, C. H. Papadimitriou, M. Schapira, Y. Singer, and C. Umans. Inapproximability for vcg-based combinatorial auctions. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010a.

D. Buchfuhrer, M. Schapira, and Y. Singer. Computation and incentives in combinatorial public projects. In *Proc. of the ACM conference on Electronic commerce*, pages 33–42, 2010b.

R. Day and S. Raghavan. Assignment preferences and combinatorial auctions. Working paper, University of Connecticut, April 2006.

S. Dobzinski. Two randomized mechanisms for combinatorial auctions. In *APPROX*, 2007.

S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. STOC '05, pages 610–618, 2005.

S. Dobzinski, N. Nisan, and M. Schapira. Truthful randomized mechanisms for combinatorial auctions. STOC '06, pages 644–652, 2006.

U. Feige. On maximizing welfare when utility functions are subadditive. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 41–50, 2006.

G. Goel, C. Karande, P. Tripathi, and L. Wang. Approximability of combinatorial problems with multi-agent submodular cost functions. In *Proceedings of the 50th Annual Symposium on Foundations of Computer Science*, 2009.

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

S. Iwata and K. Nagano. Submodular function minimization under covering constraints. In *Proceedings of the 50th Annual Symposium on Foundations of Computer Science*, 2009.

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.

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

S. Lahaie and D. C. Parkes. Applying learning algorithms to preference elicitation. In *ACM Conference on Electronic Commerce*, pages 180–188, 2004.

S. Lahaie, F. Constantin, and D. Parkes. More on the power of demand queries in combinatorial auctions: learning atomic languages and handling incentives. In *Proceedings of the 19th international joint conference on Artificial intelligence*, pages 959–964, 2005.

B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *ACM Conference on Electronic Commerce*, pages 18–28, 2001.

K. Murota. *Discrete Convex Analysis*. SIAM, 2003.

N. Nisan. Chapter 9: Bidding Languages for Combinatorial Auctions . In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*. MIT Press, 2006.

N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge, 2007.

T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, pages 542–547, 2001.

Y. Singer. Budget feasible mechanisms. FOCS '10, pages 765–774, 2010.

Z. Svitkina and L. Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. In *Proceedings of the 49th Annual IEEE Symposium onFoundations of Computer Science*, 2008.

D. Vainsencher, O. Dekel, and S. Mannor. Bundle selling by online estimation of valuation functions. In *ICML*, 2011.

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

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

Appendix A. Additional proofs for section 3.1

Lemma 1 *For large enough n, there exist $A_1, ..., A_k \subseteq [n]$ where $k = n^{\frac{1}{3}\log\log n}$ such that:*
(i) $\sqrt{n}/2 \leq |A_i| \leq 2\sqrt{n}$ for any $1 \leq i \leq k$;
(ii) $|A_i \cap A_j| \leq \log n$ for any $1 \leq i < j \leq k$.

Proof Let random variables $Y_i = |A_i|$ and $X_{ij} = |A_i \cap A_j|$. Obviously, $E[Y_i] = \sqrt{n}$ and $E[X_{ij}] = 1$. By Chernoff bounds, we have

$$\Pr\left[\, \sqrt{n}/2 < Y_i < 2\sqrt{n} \,\right] > 1 - 2e^{-\sqrt{n}/8}$$

and

$$\Pr\left[\, X_{ij} > \ln n \,\right] < \tfrac{e^{\ln n}}{\ln n^{\ln n}} = n^{-(\ln\ln n - 1)}, \;\; \forall 1 \le i < j \le k.$$

By union bound the probability that (i) and (ii) hold is at least $1 - 2\,k\,e^{-\sqrt{n}/8} - k^2 n^{-(\ln\ln n - 1)} > 0$. ■

Lemma 2 *Any fractionally subadditive valuation f satisfies*

$$f(T) = \max\{\sum_{i \in T} x_i | x \in P(f)\} \quad \text{for} \quad \text{any} \;\; T \subseteq [n],$$

where $P(f)$ is the associated polyhedron $\{x \in \mathbf{R}_+^n : \sum_{i \in S} x_i \le f(S), \forall S \subseteq [n]\}$.

Proof The proof of this result involves a pair of dual linear programs, one corresponding to the maximization and another one that is tailored for fractional subadditivity, with an optimal objective value of $f(T)$. Formally, for any $T \subseteq [n]$ we have $\sum_{i \in T} x_i \le f(T)$ for any $x \in P(f)$. Therefore $f(T) \ge \max\{\sum_{i \in T} x_i | x \in P(f)\}$. Now we prove that in fact $f(T) \le \max\{\sum_{i \in T} x_i | x \in P(f)\}$. Consider the linear programming (LP1) for the quantity $\max\{x(T) | x \in P(f)\}$ and its dual (LP2): we assign a dual variable y_S for each constraint in (LP1), and we have a constraint corresponding to each primal variable indicating that the total amount of dual corresponding to a primal variable should not exceed its coefficient in the primal objective.

$$\begin{aligned} \max \sum_{i \in T} x_i & \qquad \text{(LP1)} \\ s.t. \sum_{i \in S} x_i \le f(S) & \qquad \forall S \subseteq [n], \\ x_i \ge 0 & \qquad \forall i \in [n]. \end{aligned} \qquad\qquad \begin{aligned} \min \sum_{S \subseteq [n]} y_S f(S) & \qquad \text{(LP2)} \\ s.t. \sum_{S : i \in S} y_S \ge 1 & \qquad \forall i \in T, \\ y_S \ge 0 & \qquad \forall S \subseteq [n]. \end{aligned}$$

The classical theory of linear optimization gives that the optimal primal solution equals the optimal dual solution. Let y^* be an optimal solution of (LP2). Therefore

$$\sum_{S \subseteq [n]} y_S^* f(S) = \max\{x(T) | x \in P(f)\}.$$

Since f is fractionally subadditive and $\sum_{S : i \in S} y_S^* \ge 1, \forall i \in T$, we have $f(T) \le \sum_{S \subseteq [n]} y_S^* f(S)$, hence $f(T) \le \max\{x(T) | x \in P(f)\}$. Thus $f(T) = \max\{\sum_{i \in T} x_i | x \in P(f)\}$. ■

A.1. Additional details for the proof of Theorem 1

For PMAC-learning XOS valuations to an $\sqrt{n + \varepsilon}$ factor, we apply Algorithm 1 with parameters $R = n$, ϵ, and $p = 2$. The proof of correctness of Algorithm 1 follows by using the structural result in Claim 1 and a technique of Balcan and Harvey (2011). For completeness, we provide here the full details of this proof (the description is taken from (Balcan and Harvey, 2011)).

Because of the multiplicative error allowed by the PMAC-learning model, we separately analyze the subset of the instance space where f^* is zero and the subset of the instance space where f^* is non-zero. For convenience, we define:

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

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 (Kearns and Vazirani, 1994; Vapnik, 1998) 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 $c^* : \mathbb{R}^m \to \{-1, +1\}$, $c^*(x) = \operatorname{sgn}(u^\mathsf{T} x)$. The examples induced by D' and c^* are called *linearly separable* since there exists a vector u such that $c^*(x) = \operatorname{sgn}(u^\mathsf{T} x)$. 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 for each. The sample from D' is

$$\begin{aligned} (\chi(S), (f^*(S))^2) \qquad &\text{(if the coin is heads)} \\ (\chi(S), (n+\varepsilon)\cdot(f^*(S))^2) \qquad &\text{(if the coin is tails).} \end{aligned}$$

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 the assumption that for the linear function $f(S) = \hat{w}^\mathsf{T}\chi(S)$ with $w \in \mathbb{R}^n$, we have $(f^*(S))^2 \leq \hat{f}(S) \leq n(f^*(S))^2$ for all $S \subseteq [n]$. Let $u = ((n + \varepsilon/2)\cdot w, -1) \in \mathbb{R}^m$. For any point x in the support of D' we have

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

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,

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

Our algorithm is as follows. It first partitions the training set $\mathcal{S} = \{(S_1, f^*(S_1)), \ldots, (S_m, f^*(S_m))\}$ 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

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

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 $x_i = (\chi(A_i), (f^*(A_i))^2)$; otherwise set $x_i = (\chi(A_i), (n + \varepsilon) \cdot (f^*(A_i))^2)$. The last step of our algorithm is to solve a linear program in order to find a linear separator $u = (\hat{w}, -z)$ where $\hat{w} \in \mathbb{R}^n$, $z \in \mathbb{R}$ 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 $f(S) = (\frac{1}{(n+\varepsilon)z}\hat{w}^{\mathsf{T}}\chi(S))^{1/2}$.

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 $(\chi(S), (f^*(S))^2)$ and $(\chi(S), (n+\varepsilon) \cdot (f^*(S))^2)$ 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 $f(S) = (\frac{1}{(n+\varepsilon)z}\hat{w}^{\mathsf{T}}\chi(S))^{1/2}$ approximates f^* to within a factor $n + \varepsilon$ on all the points in the set $\mathcal{Y}$. To see this notice that for any point $S \in \mathcal{Y}$, we have

$$\hat{w}^{\mathsf{T}}\chi(S) - z(f^*(S))^2 > 0 \qquad \text{and} \qquad \hat{w}^{\mathsf{T}}\chi(S) - z(n+\varepsilon)(f^*(S))^2 < 0$$
$$\implies \quad \frac{1}{(n+\varepsilon)z}\hat{w}^{\mathsf{T}}\chi(S) \;<\; (f^*(S))^2 \;<\; (n+\varepsilon)\frac{1}{(n+\varepsilon)z}\hat{w}^{\mathsf{T}}\chi(S).$$

So, for any point in $S \in \mathcal{Y}$, the function $f(S)^2 = \frac{1}{(n+\varepsilon)z}\hat{w}^{\mathsf{T}}\chi(S)$ approximates $(f^*(\cdot))^2$ to within a factor $n + \varepsilon$. Moreover, by design the function f correctly labels as 0 all the examples in $\mathcal{P}_0$. To finish the proof, we now note two important facts: for our choice of $m = \frac{16n}{\epsilon}\log\left(\frac{n}{\delta\epsilon}\right)$, with high probability both $\mathcal{P} \setminus \mathcal{Y}$ and $\mathcal{Z} \setminus \mathcal{P}_0$ have small measure.

Claim 1 *With probability at least $1 - \delta$, the set $\mathcal{Z} \setminus \mathcal{P}_0$ has measure at most ϵ.*

Proof Let $\mathcal{P}_k = \{\, S : S \subseteq \mathcal{U}_k \,\}$. Suppose that, for some k, the set $\mathcal{Z} \setminus \mathcal{P}_k$ 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 $\mathcal{Z} \setminus \mathcal{P}_k$ is at most $(1-\epsilon)^{\log(n/\delta)/\epsilon} \leq \delta/n$. On the other hand, if one of them does lie in $\mathcal{Z} \setminus \mathcal{P}_k$, 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 $m \geq n\log(n/\delta)/\epsilon$, with probability at least δ the set $\mathcal{Z} \setminus \mathcal{P}_m$ has measure at most ϵ. ■

We now prove:

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

Proof [Proof of Claim 2] 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 $\mu = \mathbf{E}[\, a \,] = qm$. 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 (Vapnik, 1998) 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 $(\chi(S), (f^*(S))^2)$ or $(\chi(S), (n+\varepsilon) \cdot (f^*(S))^2)$ is incorrectly labeled, given that $S \in \mathcal{P}$, is at most $2\epsilon/q$. Thus

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

as required. ■

In summary, our algorithm outputs a hypothesis f approximating f^* to within a factor $(n+\varepsilon)^{1/2}$ on $\mathcal{Y} \cup \mathcal{P}_m$. The complement of this set is $(\mathcal{Z} \setminus \mathcal{P}_0) \cup (\mathcal{P} \setminus \mathcal{Y})$, which has measure at most 3ϵ, with probability at least $1 - 3\delta$.

Appendix B. Additional result concerning learnability of OXS valuations

In this section we provide additional details concerning the learnability of OXS valuations, focusing on interesting subclasses of OXS functions that arise in practice, namely OXS functions representable with a small number of MAX trees or leaves.

Motivating Examples For example, a traveler presented with a collection of plane tickets, hotel rooms, and rental cars for a given location might value the bundle as the sum of his values on the best ticket, the best hotel room, and the best rental car. This valuation is OXS, with one MAX tree for each travel requirement. The number of MAX trees, i.e. travel requirements, is small but the number of leaves in each tree may be large. As another example, consider for example a company producing airplanes that must procure many different components for assembling an airplane. The number of suppliers for each component is small, but the number of components may be very large (more than a million in today's airplanes). The company's value for a set of components of the same type, each from a different supplier, is its highest value for any such component. The company's value for a set of components of different types is the sum of the values for each type. This valuation is representable as an OXS, with one tree for each component type. The number of leaves, i.e. suppliers, in each MAX tree is small but there may be many such trees. In this section, we show good PMAC-learning guarantees for classes of functions of these types. Formally:

B.1. Additional results for Theorem 4

We prove that Algorithm 2 can be used to PAC-learn (i.e. PMAC-learn with $\alpha = 1$) any unit-demand valuation.

Lemma 2 *The class of unit-demand valuations is properly PAC-learnable by using* $m = O(n \ln(n/\delta)/\varepsilon)$ *training examples and time* $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$.

Proof We first show how to solve the consistency problem in polynomial time: given a sample $(S_1, f^*(S_1)), \ldots, (S_m, f^*(S_m))$ we show how to construct in polynomial time a unit-demand function f that is consistent with the sample, i.e., $f(S_l) = f^*(S_l)$, for $l \in \{1, 2, \ldots, m\}$. In particular, using the reasoning in Theorem 3 for $R = 1$, we show that the unit-demand hypothesis f output by Algorithm 2 is consistent with the samples. We have

$$f(S_l) = \max_{i \in S_l} f(i) = \max_{i \in S_l} \min_{j : i \in S_j} f^*(S_j) \leq f^*(S_l).$$

Also note that for any $i \in S_l$ we have $f^*(i) \leq f^*(S_l)$, for $l \in \{1, 2, \ldots, m\}$. So $f(i) \geq f^*(i) \quad$ for any $i \in S_1 \cup \ldots \cup S_m$.. Therefore for any $l \in \{1, 2, \ldots, m\}$ we have :

$$f^*(S_l) = \max_{i \in S_l} f^*(i) \leq \max_{i \in S_l} f(i) = f(S_l).$$

Thus $f^*(S_l) = f(S_l)$ for $l \in \{1, 2, \ldots, m\}$.

We now claim that $m = O(n \ln(n/\delta)/\varepsilon)$ training examples are sufficient so that with probability at least $1 - \delta$, the hypothesis f produced has error at most ϵ. In particular, notice that Algorithm 2 guarantees that $f(i) \in \{f^*(1), \ldots, f^*(n)\}$ for all i. This means that for any given target function f^*, there are at most n^n different possible hypotheses f that Algorithm 2 could generate. By the union bound, the probability that the algorithm outputs one of error greater than ε is at most $n^n(1-\varepsilon)^m$ which is at most δ for our given choice of m. ■

Lemma 3 *If f^* is* OXS *with at most R trees, then it is also unit-demand with at most n^R leaves (with R-tuples as items). For constant R, the family $\mathcal{F}$ of* OXS *functions with at most R trees is PAC-learnable using $O(Rn^R \log(n/\delta)/\varepsilon)$ training examples and time* $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$.

Proof We start by noting that since f^* is an OXS function representable with at most R MAX trees, then f^* is uniquely determined by its values on sets of size up to R. Formally,

$$f^*(S) = \max_{\substack{S^R \subseteq S \\ |S^R| \leq R}} f^*(S^R), \forall S \subseteq \{1, \ldots, n\} \tag{7}$$

We construct a unit-demand f', closely related to f, on meta-items corresponding to each of the $O(n^R)$ sets of at most R items. In particular, we define one meta-item i_{S^R} to represent each set $S^R \subseteq \{1, \ldots, n\}$ of size at most R and let

$$f'(i_{S^R}) = f^*(S^R).$$

We define f' as unit-demand over meta-items; i.e. beyond singleton sets, we have

$$f'(\{i_{S_1^R}, \ldots, i_{S_L^R}\}) = \max_{l=1,\ldots,L} f'(i_{S_l^R}). \tag{8}$$

By equations (7) and (8), for all S we have $f^*(S) = f'(I_S)$ where $I_S = \{i_{S^R} : S^R \subseteq S\}$.

Since we can perform the mapping from sets S to their corresponding sets I_S over meta-items in time $O(n^R)$, this implies that to PAC-learn f^*, we can simply PAC-learn f' over the $O(n^R)$ meta-items using Algorithm 2. Lemma 2 guarantees that this will PAC-learn using $O(n^R \log(n^R/\delta)/\varepsilon)$ training examples and running time $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$ for constant R. ■

Appendix C. Learnability everywhere with value queries

We also consider approximate learning with value queries (considered previously for submodular functions by Goemans et al. (2009) and Svitkina and Fleischer (2008)). This is relevant for settings where instead of passively observing the values of f^* on sets S drawn from a distribution, the learner is able to actively query the value $f^*(S)$ on sets S of its choice and the goal is to approximate with certainty the target f^* on all 2^n sets after querying the values of f^* on polynomially many sets. Formally:

Definition 5 *We say that an algorithm $\mathcal{A}$ learns the valuation family $\mathcal{F}$ everywhere with value queries with an approximation factor of $\alpha \geq 1$ if, for any target function $f^* \in \mathcal{F}$, after querying the values of f^* on polynomially (in n) many sets, $\mathcal{A}$ outputs in time polynomial in n a function f such that $f(S) \leq f^*(S) \leq \alpha f(S), \forall S \subseteq \{1, \ldots, n\}$.*

Goemans et al. (2009) show that for submodular functions the learnability factor with value queries is $\tilde{\Theta}(n^{1/2})$. We show here that their lower bound applies to the more restricted OXS and GS classes (their upper bound automatically applies). We also show that this lower bound can be circumvented for the interesting subclasses of OXS and XOS that we considered earlier (for PMAC learning), efficiently achieving a factor of R.

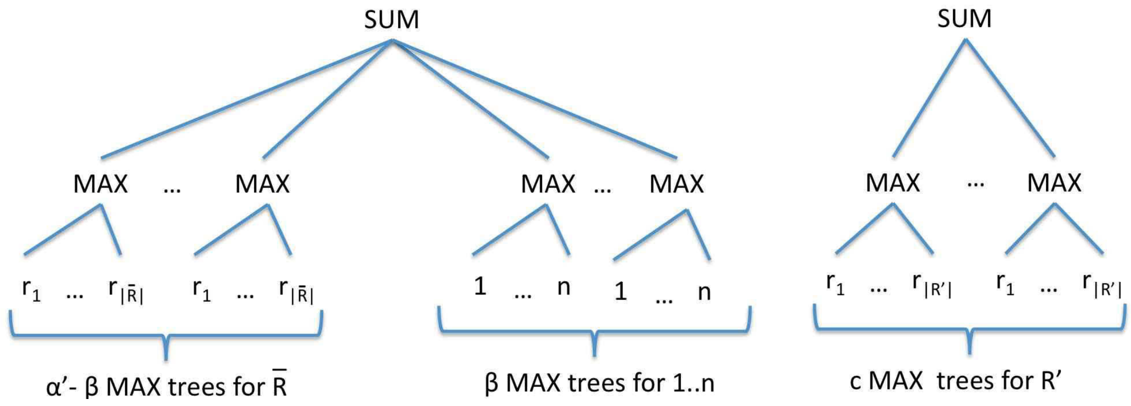
Theorem 5 ***(1)*** *The classes of* OXS *and* GS *functions are learnable with value queries with an approximation factor of $\tilde{\Theta}(n^{1/2})$.* ***(2)*** *The following classes are learnable with value queries with an approximation factor of R:* OXS *with at most R leaves in each tree,* OXS *with at most R trees,* XOS *with at most R leaves in each tree, and* XOS *with at most R trees.*

Proof [Proof Sketch] **(1)** We show in Appendix D that the family of valuation functions used in (Goemans et al., 2009) for proving the $\Omega(\frac{n^{1/2}}{\log n})$ lower bound for learning submodular valuations with value queries is contained in OXS. The valuations in this family are of the form $g_{23}(S) = \min(|S|, \alpha')$ and $g^R(S) = \min(\beta + |S \cap ((\{1, \ldots, n\}) \setminus R)|, |S|, \alpha')$ for $\alpha' = xn^{1/2}/5, \beta = x^2/5$ with $x^2 = \omega(\log n)$ and R a subset of $\{1, \ldots, n\}$ of size α' (chosen uniformly at random). These valuations are OXS; for example, $g_{23}(S)$ can be expressed as a SUM of α' MAX trees, each having as leaves all items in $[n]$ with weight 1.

(2) To establish learnability for these interesting subclasses, we recall that for the first three of them (Theorems 3 and 4) any valuation f^* in each class was approximated to an R factor by a function f that only depended on the values of f^* on items. An analogous result holds for the fourth class, i.e. XOS with at most R trees – indeed, for such an XOS f^*, we have $\frac{1}{R}\sum_{i \in S} f^*(\{i\}) \leq f^*(S) \leq R\frac{1}{R}\sum_{i \in S} f^*(\{i\}), \forall S \subseteq [n]$. One can then query these n values and output the corresponding valuation f. ■

Note: We note that the lower bound technique in Goemans et al. (2009) has been later used in a sequence of papers (Iwata and Nagano, 2009; Goel et al., 2009; Svitkina and Fleischer, 2008) concerning optimization under submodular cost functions and our result (Lemma 5 in particular) implies that all the lower bounds in these papers apply to the smaller class of OXS functions.

Note: We also note that since the class of XOS valuations contains all submodular valuations, the lower bound of Goemans et al. (2009) implies that the XOS class is not learnable everywhere with value queries to a $o(\frac{n^{1/2}}{\log n}) = \tilde{o}(n^{1/2})$ factor. For the same $\Omega(\frac{n^{1/2}}{\log n})$ lower bound, our proof technique (and associated family of XOS valuations) for Theorem 1 offers a simpler argument than that in Goemans et al. (2009).



(*a*) OXS representation for the function in Eq. (9) when $n > \alpha' > \beta$. (*b*) OXS representation for the function in Eq. (10).

Figure 1: OXS representations for functions in Appendix D. All leaves have value 1.

Appendix D. Additional results for Theorem 5

Goemans et al. (2009) proved that a certain matroid rank function $f_{R,\alpha',\beta}(\cdot)$, defined below, is hard to learn everywhere with value queries to an approximation factor of $o(\sqrt{\frac{n}{\ln n}})$. We show that the rank function $f_{R,\alpha',\beta}(\cdot)$ is in OXS (all leaves in all OXS trees will have value 1). $f_{R,\alpha',\beta}(\cdot) : 2^{\{1,\ldots,n\}} \to \mathbb{R}$ is defined as follows. Let subset $R \subseteq \{1,\ldots,n\}$ and $\bar{R} = (\{1,\ldots,n\}) \setminus R$ its complement. Also fix integers $\alpha', \beta \in \mathbb{N}$. Then

$$f_{R,\alpha',\beta}(S) = \min(\beta + |S \cap \bar{R}|, |S|, \alpha'),\ \forall S \subseteq \{1,\ldots,n\} \tag{9}$$

As a warm-up, we show that a simpler function than $f_{R,\alpha',\beta}(\cdot)$ is in OXS. This simpler function essentially corresponds to $\beta = 0$ and will be used in the case analysis for establishing that $f_{R,\alpha',\beta}(\cdot)$ is in OXS.

Lemma 4 *Let $R' \subseteq \{1,\ldots,n\}$ and $c \in \mathbb{N}$. Then the function $f(\cdot) : 2^{\{1,\ldots,n\}} \to \mathbb{R}$ defined as*

$$f_{R',c}(S) = \min(c, |S \cap R'|), \forall S \subseteq \{1,\ldots,n\} \tag{10}$$

is in OXS.

Proof For ease of notation let $f(\cdot) = f_{R',c}(\cdot)$. We assume $c < |R'|$; otherwise, $f(S) = |S \cap R'|, \forall S \subseteq \{1,\ldots,n\}$, which is a linear function ($f(S) = \sum_{x \in S} f(x)$ where $f(x) = 1$ if $x \in R'$ and $f(x) = 0$ otherwise) and any linear function belongs to the class OXS (Lehmann et al., 2001). Assuming $c < |R'|$, we construct an OXS tree T with c MAX trees, each with one leaf for every element in R'. All leaves have value 1. We refer the reader to Fig. 1(*b*).

Then $T(S) = f(S), \forall S \subseteq \{1,\ldots,n\}$ since $f(S)$ represents the smaller of the number of elements in $S \cap R'$ (that can each be taken from a different MAX tree in T) and c. Note that $T(S) \leq c, \forall S \subseteq \{1,\ldots,n\}$. ■

Lemma 5 *The matroid rank function $f_{R,\alpha',\beta}(\cdot)$ is in the class* OXS.

Proof For ease of notation let $f(\cdot) = f_{R,\alpha',\beta}(\cdot)$. If $n \leq \alpha'$ then[10] $|S| \leq \alpha', \forall S \subseteq \{1, \ldots, n\}$ and

$$f(S) = \min(\beta + |S \cap \bar{R}|, |S|) = |S \cap \bar{R}| + \min(\beta, |S \cap R|) \tag{11}$$

From the proof of Lemma 4 we get that the function $f'(S) = \min(\beta, |S \cap R|)$ has an OXS tree T' (i.e. $T'(S) = f'(S), \forall S \subseteq \{1, \ldots, n\}$) with β MAX trees each with leaves only in R. We can create a new tree T by adding $|\bar{R}|$ MAX trees to T', each with one leaf for every element in $\bar{R}$, and we get $T(S) = f(S), \forall S \subseteq \{1, \ldots, n\}$. The additional $|\bar{R}|$ MAX trees encode the $|S \cap \bar{R}|$ term in Eq. (11). If $\alpha' \leq \beta$ then $f(S) = \min(\alpha', |S|)$; the claim follows by Lemma 4 for $c = \alpha', R' = \{1, \ldots, n\}$. We can thus assume that $n > \alpha' > \beta$. We prove that the OXS tree T, containing the two types of MAX trees below, represents f, i.e. $T(S) = f(S), \forall S$. We refer the reader to Fig. 1(*a*).

- $\alpha' - \beta$ MAX trees $T_1 \ldots T_{\alpha'-\beta}$, each having as leaves all the elements in $\bar{R}$ with value 1.
- β MAX trees $T_{\alpha'-\beta+1} \ldots T_{\alpha'}$, each having as leaves all the elements (in $\{1, \ldots, n\}$) with value 1.

We note that $T(S) \leq \min(|S|, \alpha')$ as no set S can use more than $|S|$ leaves and T has exactly α' trees. We distinguish the following cases

- $|S| \leq \beta$ implying $f(S) = |S|$ and $T(S) = |S|$ as $|S|$ leaves can be taken each from $|S|$ trees in $T_{\alpha'-\beta+1} \ldots T_{\alpha'}$.
- $f(S) = \alpha' \leq \min(\beta + |S \cap \bar{R}|, |S|)$. We claim $T(S) \geq \alpha'$. There must exist $\alpha' - \beta$ elements in $|S \cap \bar{R}|$, that we can select one from each tree $T_1 \ldots T_{\alpha'-\beta}$. Also $|S| \geq \alpha'$ and we can take the remaining β elements from $T_{\alpha'-\beta+1} \ldots T_{\alpha'}$.
- $f(S) = |S| \leq \min(\beta + |S \cap \bar{R}|, \alpha')$. This implies $|S \cap R| \leq \beta$ and $|S| \leq \alpha'$. We claim $T(S) \geq |S|$: we can take all needed elements in $S \cap \bar{R}$ from $T_1 \ldots T_{\alpha'-\beta}$ (and from $T_{\alpha'-\beta+1} \ldots T_{\alpha'}$ if $|S \cap \bar{R}| > \alpha' - \beta$) and elements in $S \cap R$ from $T_{\alpha'-\beta+1} \ldots T_{\alpha'}$.
- $f(S) = \beta + |S \cap \bar{R}| \leq \min(|S|, \alpha')$. We claim $T(S) \geq \beta + |S \cap \bar{R}|$: we can take $\beta \leq |S \cap R|$ elements from $T_{\alpha'-\beta+1} \ldots T_{\alpha'}$ and $|S \cap \bar{R}| \leq \alpha' - \beta$ elements from $T_1 \ldots T_{\alpha'-\beta}$. Finally, $T(S) \leq \beta + |S \cap \bar{R}|$ since at most all elements in $S \cap \bar{R}$ can be taken from $T_1 \ldots T_{\alpha'-\beta}$ and at most β elements in $S \cap R$ from $T_{\alpha'-\beta+1} \ldots T_{\alpha'}$.

■

Appendix E. Learning with prices

We introduce here a new paradigm that is natural in many applications where the learner can repeatedly obtain information on the unknown valuation function of an agent via the agent's *decisions to purchase or not* rather than via random samples from this valuation or via queries to it. In this framework, the learner does not obtain the value of f^* on each input set $S_1, S_2, \ldots$. Instead, for each input set S_l, the learner observes S_l, quotes a price p_l (of its choosing) on S_l and obtains one

[10] We note that $n > \alpha'$ in Goemans et al. (2009). We consider this case for completeness.

bit of information: whether the agent purchases S_l or not, i.e. whether $p_l \leq f^*(S_l)$ or not. The goal remains to approximate the function f^* well, i.e. within an α multiplicative factor: on most sets from D with high confidence for PMAC-learning and on all sets with certainty for learning everywhere with value queries. The learner's challenge is in choosing prices that allow discovery of the agent valuation. This framework is a special case of demand queries (Nisan et al., 2007), where prices are: p_l on S_l and ∞ elsewhere. We call PMAC-learning with prices and VQ-learning with prices the variants of this framework applied to PMAC and learning everywhere with value queries models. Each variant in this framework offers less information to the learner than its respective basic model.

Clearly, all our PMAC-learning lower bounds still hold for PMAC-learning with prices. More interestingly, our upper bounds still hold as well. In particular, we provide a reduction from the problem of PMAC-learning with prices to the problem of learning a linear separator, for functions f^* such that for some $p > 0$, $(f^*)^p$ can be approximated to a β factor by a linear function. Such f^* can be PMAC-learned to a $\beta^{1/p}$ factor by Algorithm 1. What we show in Theorem 6 below is that such f^* are PMAC-learnable with prices to a factor of $(1 + o(1))\beta^{1/p}$ using only a small increase in the number of samples over that used for PMAC learning.

For convenience, we assume that all valuations are integral and that H is an upper bound on the values of f^*, i.e. $f^*(S) \leq H, \forall S \subseteq [n]$.

Theorem 6 *Consider a family $\mathcal{F}$ of valuations s. t. the p-th power of any $f^* \in \mathcal{F}$ can be approximated to a β factor by a linear function: i.e., for some w we have $w^\mathsf{T}\chi(S) \leq (f^*(S))^p \leq \beta w^\mathsf{T}\chi(S)$ for all $S \subseteq [n]$, where $\beta \geq 1, p > 0$. Then for any $0 < \eta \leq 1$, $\mathcal{F}$ is PMAC-learnable with prices to a $(1+\eta)\beta^{1/p}$ factor using $O(\frac{n \log H}{\eta\varepsilon} \ln(\frac{n \log H}{\eta\varepsilon\delta}))$ samples and time* $\mathrm{poly}(n, \frac{1}{\epsilon}, \frac{1}{\delta}, \frac{1}{\eta})$.

Proof As in Algorithm 1, the idea is to use a reduction to learning a linear separator, but where now the examples use prices (that the algorithm can choose) instead of function values (that the algorithm can no longer observe). For each input set S_l, the purchase decision amounts to a comparison between the chosen price q_l and $f^*(S_l)$. Using the result of this comparison we will construct examples, based on the prices q_l, that are always consistent with a linear separator obtained from $w^\mathsf{T}\chi(S_l)$, the linear function that approximates $(f^*)^p$. We will sample enough sets S_l and assign prices q_l to them in such a way that for sufficiently many l, the price q_l is close to $f^*(S_l)$. We then find a linear separator that has small error on the distribution induced by the price-based examples and show this yield a hypothesis $f(S)$ whose error is not much higher on the original distribution with respect to the values of the (unknown) target function.

Specifically, we take $m = O(\frac{n \log H}{\eta\varepsilon} \ln(\frac{n \log H}{\eta\varepsilon\delta}))$ samples, and for convenience define $N = \lfloor \log_{1+\eta/3} H \rfloor$. We assign to each input bundle S_l a price q_l drawn uniformly at random from $\{(1 + \eta/3)^i\}$ for $i = 0, 1, 2, \ldots, N+1$, and present bundle S_l to the agent at price q_l. The key point is that for bundles S_l such that $f^*(S_l) \geq 1$ this ensures at least a $\frac{1}{N+2}$ probability that $f^*(S_l)(1+\eta/3)^{-1} < q_l \leq f^*(S_l)$ and at least a $\frac{1}{N+2}$ probability that $f^*(S_l) < q_l \leq f^*(S_l)(1 + \eta/3)$ (the case of $f^*(S_l) = 0$ will be noticed when the agent does not purchase at price $q_l = 1$ and is handled as in the proof of Theorem 1).

We construct new examples based on these prices and purchase decisions as follows. If $f^*(S_l) < q_l$ (i.e. the agent does not buy) then we let $(x_l, y_l) = ((\chi(S_l), \beta q_l^p), -1)$. If $f^*(S_l) \geq q_l$ (i.e. the agent buys) then we let $(x_l, y_l) = ((\chi(S_l), q_l^p), +1)$. Note that by our given assumption, the examples constructed are always linearly separable. In particular the label y_l matches $\mathrm{sgn}((\beta w, -1)^\mathsf{T} x_l)$

in each case: $\beta w^{\mathsf{T}}\chi(S_l) \leq \beta(f^*(S_l))^p < \beta q_l^p$ and $q_l^p \leq (f^*(S_l))^p \leq \beta w^{\mathsf{T}}\chi(S_l)$ respectively. Let D_{buy}^{n+1} denote the induced distribution on $\mathbb{R}^{n+1}$. We now find a linear separator $(\hat{w}, -z) \in \mathbb{R}^{n+1}$, where $\hat{w} \in \mathbb{R}^n$ and $z \in \mathbb{R}_+$, that is consistent with $(x_l, y_l), \forall l$. We construct an intermediary hypothesis $f'(S) = \frac{1}{\beta z}\hat{w}^{\mathsf{T}}\chi(S)$ based on the learned linear separator. The hypothesis output will be $f(S) = \frac{1}{1+\eta/3}(f'(S))^{1/p}$.

By standard VC-dimension sample-complexity bounds, our sample size m is sufficient that the linear separator $(\hat{w}, -z)$ has error on D_{buy}^{n+1} at most $\frac{\varepsilon}{N+2}$ with probability at least $1 - \delta$. We now show that this implies that with probability at least $1 - \delta$, hypothesis $f(S)$ approximates $f^*(S)$ to a factor $(1 + \eta/3)^2\beta^{1/p} \leq (1 + \eta)\beta^{1/p}$ over D, on all but at most an ε probability mass, as desired.

Specifically, consider some bundle S for which $f(S)$ does *not* approximate $f^*(S)$ to a factor $(1 + \eta/3)^2\beta^{1/p}$ and for which $f^*(S) \geq 1$ (recall that zeroes are handled separately). We just need to show that for such bundles S, there is at least a $\frac{1}{N+2}$ probability (over the draw of price q) that $(\hat{w}, -z)$ makes a mistake on the resulting example from D_{buy}^{n+1}. There are two cases to consider:

1. It could be that f is a bad approximation because $f(S) > f^*(S)$. This implies that $f'(S) > [(1 + \eta/3)f^*(S)]^p$ or equivalently that $\hat{w}^{\mathsf{T}}\chi(S) > \beta z[(1 + \eta/3)f^*(S)]^p$. In this case we use the fact that there is a $\frac{1}{N+2}$ chance that $f^*(S) < q \leq f^*(S)(1+\eta/3)$. If this occurs, then the agent doesn't buy (yielding $x = (\chi(S), \beta q^p), y = -1$) and yet $\hat{w}^{\mathsf{T}}\chi(S) > \beta z q^p$. Thus the separator mistakenly predicts positive.

2. Alternatively, it could be that $(1 + \eta/3)^2\beta^{1/p}f(S) < f^*(S)$. This then implies that $(1 + \eta/3)\beta^{1/p}f'(S)^{1/p} < f^*(S)$ or equivalently that $\hat{w}^{\mathsf{T}}\chi(S) < z(\frac{f^*(S)}{1+\eta/3})^p$. In this case, we use the fact that there is a $\frac{1}{N+2}$ chance that $\frac{f^*(S)}{1+\eta/3} < q \leq f^*(S)$. If this occurs, then the agent does buy (yielding $x = (\chi(S), q^p), y = +1$) and yet $\hat{w}^{\mathsf{T}}\chi(S) < zq^p$. Thus the separator mistakenly predicts negative.

Thus, the error rate under D_{buy}^{n+1} is at least a $\frac{1}{N+2}$ fraction of the error rate under D, and so a low error under D_{buy}^{n+1} implies a low error under D as desired. ■

Note: We note that if there is an underlying desired pricing algorithm $\mathcal{A}$, for each input set S_l we can take the price of S_l to be $\mathcal{A}(S_l)$ with probability $1 - \tilde{\varepsilon}$ and a uniformly at random price in $\{1, 2, 4 \ldots, H/2, H\}$ as in the previous result with probability $\tilde{\varepsilon}$. We can also recover our upper bounds on learnability everywhere with value queries (the corresponding lower bounds clearly hold).

We can also recover our upper bounds on learnability everywhere with value queries (the corresponding lower bounds clearly hold). By sequentially setting prices $1, 2, 4 \ldots, H/2, H$ on each item we can learn f^*'s values on items within a factor of 2. Our structural results proving the approximability of f^* from interesting classes with a function that only depends on $f^*(\{1\}), \ldots, f^*(\{n\})$ then yield the VQ-learnability with prices of these classes.

Theorem 7 *The following classes are VQ-learnable with prices to within an $2R$ factor:* OXS *with at most R trees,* OXS *with at most R leaves in each tree,* XOS *with at most R trees, and* XOS *with at most R leaves in each tree.*