

# Learning Valuation Functions

Maria Florina Balcan\*    Florin Constantin†    Satoru Iwata‡    Lei Wang§

## Abstract

A core element of microeconomics and game theory is that consumers have valuation functions over bundles of goods and that these valuation functions drive their purchases. In particular, the value assigned to a bundle need not be the sum of values on the individual items but rather is often a more complex function of how the items relate. The literature considers a hierarchy of valuation classes that includes subadditive, XOS (i.e. fractionally subadditive), submodular, and OXS valuations. Typically it is assumed that these valuations are known to the center or that they come from a known distribution. Two recent lines of work, by Goemans et al. (SODA 2009) and by Balcan and Harvey (STOC 2011), have considered a more realistic setting in which valuations are learned from data, focusing specifically on submodular functions.

In this paper we consider the approximate learnability of valuation functions at all levels in the hierarchy. We first study their learnability in the distributional learning (PAC-style) setting due to Balcan and Harvey (STOC 2011). We provide nearly tight lower and upper bounds of  $\tilde{\Theta}(n^{1/2})$  on the approximation factor for learning XOS and subadditive valuations, both important classes that are strictly more general than submodular valuations. Interestingly, we show that the  $\tilde{\Theta}(n^{1/2})$  lower bound can be circumvented for XOS functions of polynomial complexity; we provide an algorithm for learning the class of XOS valuations with a representation of polynomial size to within an  $O(n^\epsilon)$  approximation factor in running time  $n^{O(1/\epsilon)}$  for any  $\epsilon > 0$ . We also establish learnability and hardness results for subclasses of the class of submodular valuations, i.e. gross substitutes valuations and interesting subclasses of OXS valuations.

In proving our results for the distributional learning setting, we provide novel structural results for all these classes of valuations. We show the implications of these results for the learning everywhere with value queries model, considered by Goemans et al. (SODA 2009).

Finally, we also introduce a more realistic variation of these models for economic settings, in which information on the value of a bundle  $S$  of goods can only be inferred based on whether  $S$  is purchased or not at a specific price. We provide lower and upper bounds for learning both in the distributional setting and with value queries.

---

\*Georgia Institute of Technology, ninamf@cc.gatech.edu

†Georgia Institute of Technology, florin@cc.gatech.edu

‡Kyoto University, iwata@kurims.kyoto-u.ac.jp

§Georgia Institute of Technology, leiwang2007@gatech.edu

# 1 Introduction

A central problem in commerce is understanding one’s customers. Whether for assigning prices to goods, for deciding how to bundle products, or for estimating how much inventory to carry, it is critical for a company to understand its customers’ preferences. In Economics and 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 to the company, 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<sup>1</sup>.

Motivated by such scenarios, in this paper we investigate the learnability of classes of functions commonly used to model consumers’ valuations. In particular, we focus on a wide class of 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 — we henceforth use the standard optimization terminology *subadditive valuations*. We provide upper and lower bounds on the learnability of valuation classes in a popular hierarchy [18, 24, 28, 29], with submodular functions (the only class with similar extant results [5, 17]) halfway in the hierarchy:

$$OXS \subsetneq \text{gross substitutes} \subsetneq \text{submodular (the only related learnability results [5, 17])} \subsetneq XOS \subsetneq \text{subadditive}$$

We analyze the learnability of these classes in the natural PMAC model [5] for approximate distributional learning. In this model, a learning algorithm is given a collection  $\mathcal{S} = \{S_1, \dots, S_m\}$  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  $D$ . More formally, we want:

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

for an algorithm that uses  $m = \text{poly}(n, \frac{1}{\epsilon}, \frac{1}{\delta})$  samples and that runs in  $\text{poly}(n, \frac{1}{\epsilon}, \frac{1}{\delta})$  time. In contrast, the classical PAC model [32] requires predicting *exactly* (i.e.  $\alpha = 1$ ) with high probability the values of  $f^*$  over most sets with respect to  $D$ . Thus the PMAC model can be viewed as an approximation-algorithms extension of the traditional PAC model.

Our main results in the PMAC-learning model are for superclasses of submodular valuations, namely subadditive valuations and XOS [13, 14, 24] (also known as fractionally subadditive [6, 15]) valuations. A XOS valuation represents a set of alternatives (e.g. travel destinations), where the valuation for subsets of goods (e.g. attractions) within each alternative is additive. The value of any set of goods, e.g. dining and skiing, is the highest value for these goods among all alternatives. That is, an XOS valuation is essentially a depth-two tree with a MAX root over SUM trees with goods as leaves. XOS valuations are intuitive and very expressive: they can represent any submodular valuation [24] and can approximate any valuation in the subadditive superclass to a  $O(\log n)$  factor [6, 12]. We also consider subclasses of submodular functions in the hierarchy, namely gross substitutes [10, 18] and OXS [9, 11, 14, 30] functions. Gross substitutes valuations are characterized by the lack of pairwise synergies among items: for example, if the value of each of three items is the same, then no pair can have a strictly higher value than the other two pairs. Finally, the OXS class includes valuations representable as the SUM of MAX of item values. All these classes include linear valuations.

We also analyze the model of approximate learning everywhere with value queries, due to Goemans et al. [17]. In this model, the learner can adaptively pick a sequence of sets  $S_1, S_2, \dots$  and query the values  $f^*(S_1), f^*(S_2), \dots$ . Unlike the high confidence and high accuracy requirements of PMAC, this model requires approximately learning  $f^*(\cdot)$  with certainty on all  $2^n$  sets. We provide upper and lower bounds in this model for the same valuation classes.

Finally, we introduce a more realistic variation of these models, in which the learner can obtain information only via prices. This variation is natural in settings where an agent with valuation  $f^*$  is interested in purchases of goods.

---

<sup>1</sup>See e.g. <http://bit.ly/lS774D> for an example of an airline asking customers for a “reasonable” price for in-flight Internet.

**Our Results.** We establish lower and upper bounds, the most general of them being almost tight, on the learnability of valuation classes in the aforementioned hierarchy.

1. 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 our upper bound is to show that any XOS function can be approximated by the square root of a linear function to 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 via a number of efficient algorithms. Our  $\Omega(\sqrt{n}/\log n)$  lower bound is information theoretic, applying to any procedure that uses a polynomial number of samples. We also show an  $O(\sqrt{n} \log n)$  upper bound on the learnability of subadditive valuations in the PMAC model.
2. We establish a target-dependent learnability result for XOS functions. Namely, we show the class of XOS functions representable with at most  $R$  trees can be PMAC-learned to an  $O(R^\eta)$  factor in time  $n^{O(1/\eta)}$  for any  $\eta > 0$ . In particular, for  $R$  polynomial in  $n$ , we get learnability to an  $O(n^\eta)$  factor in time  $n^{O(1/\eta)}$  for any  $\eta > 0$ . 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  $S$ . Conceptually, this result highlights the importance of the complexity of the target function for polynomial time learning<sup>2</sup>.
3. 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 [4, 6, 8], but we are the first to study their learnability. We also show that the previous  $\tilde{\Omega}(n^{1/3})$  lower bound for PMAC-learning submodular functions [5] applies to the much simpler class of gross substitutes.
4. The structural results we derive for analyzing learnability in the distributional learning setting also have implications for the model of exact learning with value queries [7, 17, 31]. In particular, they lead to new upper bounds for XOS and OXS as well as new lower bounds for XOS, gross substitutes, and OXS.
5. Finally, we introduce a new model for learning with prices in which the learner receives less information on the values  $f^*(S_1), f^*(S_2), \dots$ : 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).

Our results are summarized in Table 1. Note that all our upper bounds are efficient and all the lower bounds are information theoretic. Our analysis has a number of interesting byproducts that should be of interest to the Combinatorial Optimization community. For example, it implies that recent lower bounds of [7, 16, 17, 31] on optimization under submodular cost functions also apply to the smaller classes of OXS and gross substitutes.

**Related Work** We study classes of valuations with fundamental properties (subadditivity and submodularity) or that are natural constructs used widely for optimization in economic settings [28]: XOS [6, 13, 14, 15, 24], i.e. MAX of SUMs, OXS [8, 9, 11, 30], i.e. SUM of MAXs, and gross substitutes, fundamental in allocation problems [2, 10, 18].

We focus on two widely studied learning paradigms: approximate learnability in a distributional setting [1, 21, 32, 33] and approximate learning everywhere with value queries [7, 17, 22, 31]. In the first paradigm, we use a model introduced by [5] for the approximate learnability of submodular functions. We circumvent the main negative result in [5] for certain interesting classes and match the main positive result in [5] for the more general XOS and subadditive classes. With a few recent exceptions [17, 31], 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 [7], polynomial or linear-threshold valuations [23] or MAX or SUM (of bundles) valuations [22]. The latter two works also consider demand queries, where the learner can specify a set of prices and obtain a preferred bundle at these prices. In contrast, in our variation of learning with prices, the learner and the agent focus on one price only (for the current bundle) instead of as many as  $2^n$  prices.

<sup>2</sup>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.

Classes of valuations	PMAC [5]	PMAC with prices	Value Queries [17]	VQ with prices
subadditive	$\Theta(n^{1/2})$ [this paper]	$\Theta(n^{1/2})$ [this paper]	$O(n)$ [folklore]	$O(n)$ [this paper]
XOS	$\Theta(n^{1/2})$ [this paper]	$\Theta(n^{1/2})$ [this paper]	$\tilde{\Omega}(n^{1/2})$ [this paper] [17]	$\tilde{\Omega}(n^{1/2})$ [this paper]
XOS with $\leq R$ trees	$O(R^\varepsilon)$ [this paper]	$O(R^\varepsilon)$ [this paper]	$O(R)$ [this paper]	$O(R)$ [this paper]
submodular	$(\tilde{\Omega}(n^{1/3}), O(n^{1/2}))$ [5]	$(\Omega(n^{1/3}), O(n^{1/2}))$ [this paper]	$\tilde{\Theta}(n^{1/2})$ [17]	–
gross substitutes	$\tilde{\Omega}(n^{1/3})$ [this paper]	$\Omega(n^{1/3})$ [this paper]	$\tilde{\Omega}(n^{1/2})$ [this paper]	$\tilde{\Omega}(n^{1/2})$ [this paper]
OXS with $\leq R$ trees or $\leq R$ leaves per tree	$O(R)$ [this paper]	$O(R)$ [this paper]	$O(R)$ [this paper]	$O(R)$ [this paper]

Table 1: Lower and upper bounds for learnability factors achievable in different models for standard classes of valuations (presented in decreasing order of generality). All the upper bounds refer to polynomial time algorithms. Our construction for the  $\tilde{\Omega}(n^{1/2}/\log n) = \tilde{\Omega}(n^{1/2})$  lower bound on learning XOS valuations with value queries is simpler than the construction for the same asymptotic lower bound of Goemans et al. [17].

**Paper structure** After defining valuation classes and our models in Section 2, we study the distributional learnability of valuation classes in decreasing order of generality. First, Section 3 presents our results on XOS and subadditive valuations, including our most general bounds, that are almost tight. Section 4 presents a hardness result for gross substitutes and positive results on several interesting subclasses of OXS. Section 5 provides positive results for most of these classes in learning with value queries. Finally, in Section 6 we show that many of our results extend to a natural framework in economic applications, even though the learner receives less information in this framework.

## 2 Preliminaries

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

**Classes of Valuation Functions.** It is often the case that valuations are quite structured in terms of representation or constraints on the values of different sets. We now define and give intuition for most valuation classes we focus on.

The following standard properties have natural interpretations in economic settings. A subadditive valuation models the lack of synergies among sets: a set’s value is at most the sum of the values of its parts. A submodular valuation models decreasing marginal returns: an item  $j$ ’s marginal value cannot go up if one expands the base set  $S$  by item  $i$ .

**Definition 1.** A valuation  $f : 2^{[n]} \rightarrow \mathbb{R}_+$  is called subadditive if and only if  $f(S \cup S') \leq f(S) + f(S'), \forall S, S' \subseteq [n]$ . A valuation  $f$  is called submodular if and only if  $f(S \cup \{i, j\}) - f(S \cup \{i\}) \leq f(S \cup \{j\}) - f(S), \forall S \subseteq [n], \forall i, j \notin S$ .

XOS is an important class of subadditive, but not necessarily submodular, valuations studied in combinatorial auctions [13, 14, 15, 24]. A valuation is XOS if and only if 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 2.** A valuation  $f$  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  $w_{ji} \geq 0, \forall j = 1 \dots k, \forall i = 1 \dots n$ .

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

As already mentioned, any submodular valuation can be expressed as XOS<sup>3</sup>. When reversing the roles of operators MAX and SUM we obtain a strict subclass of submodular valuations, called OXS,<sup>4</sup> that is also relevant to auctions [11,

<sup>3</sup>As shown in [24], any submodular  $f$  can be represented as the MAX of  $n!$  SUM trees, each with  $n$  leaves: for every permutation  $\pi$  of  $[n]$ , we build a SUM tree  $T_\pi$  with one leaf for each item  $j \in [n]$ , where item  $\pi(j)$  has weight its marginal value  $f(\{\pi(1), \dots, \pi(j-1), \pi(j)\}) - f(\{\pi(1), \dots, \pi(j-1)\})$ .

<sup>4</sup>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[27, 29].

14, 24, 30]. 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<sup>5</sup> 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 3.** A unit-demand valuation  $f$  is given by weights  $\{w_1, \dots, w_n\} \subset \mathbb{R}_+$  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,  $f(S) = \max\{f_1(S_1) + \dots + f_k(S_k) : (S_1, \dots, S_k) \text{ is a partition of } S\}, \forall S \subseteq [n]$ .

Finally, we consider *gross substitutes* (GS) valuations, of great interest in allocation problems [10, 18, 28]. Informally, an agent with a gross substitutes valuation would not buy fewer items of one type (e.g. skis) if items of another type (e.g. snowboards) became more expensive. That is, items can be substituted one for another in a certain sense, which is not the case for, e.g. skis and ski boots. See Section 4 for a formal definition and more detailed discussion.

As already mentioned, the classes of valuations we reviewed thus far form a strict hierarchy. (See [24] for examples separating these classes of valuations.)

**Lemma 1.** [24]  $\text{OXS} \subsetneq \text{gross substitutes} \subsetneq \text{submodular} \subsetneq \text{XOS} \subsetneq \text{subadditive}$ .

Only the class of submodular valuations has been studied from an approximate learning perspective [5, 17]. We study the approximate learnability of all other classes in this hierarchy, in a few natural models that we introduce now.

**Distributional Learning: PMAC.** We primarily study learning in the PMAC model of [5]. We assume that the input for a learning algorithm is 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  $f^* : 2^{[n]} \rightarrow \mathbb{R}_+$ . The goal is to output a hypothesis function  $f$  such that, with high 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 4.** We say that a family  $\mathcal{F}$  of valuations is PMAC-learnable with approximation factor  $\alpha$  if there exists an algorithm  $\mathcal{A}$  such that for any distribution  $D$  over  $2^{[n]}$ , for any target function  $f^* \in \mathcal{F}$ , and for any sufficiently small  $\varepsilon \geq 0, \delta \geq 0$ ,  $\mathcal{A}$  takes as input samples  $\{(S_i, f^*(S_i))\}_{1 \leq i \leq m}$  where each  $S_i$  is drawn independently from  $D$  and outputs a valuation  $f : 2^{[n]} \rightarrow \mathbb{R}$  such that  $\Pr_{S_1, \dots, S_m \sim D} [\Pr_{S \sim D} [f(S) \leq f^*(S) \leq \alpha f(S)] \geq 1 - \varepsilon] \geq 1 - \delta$ .  $\mathcal{A}$  must use  $m = \text{poly}(n, \frac{1}{\varepsilon}, \frac{1}{\delta})$  samples and must have running time  $\text{poly}(n, \frac{1}{\varepsilon}, \frac{1}{\delta})$ .

PMAC stands for Probably Mostly Approximately Correct (the PAC model [32] is a special case of PMAC with  $\alpha = 1$ ).

**Learning with Value Queries.** We also consider the model of learnability everywhere (in the same approximate sense) with value queries. In this model, the learning algorithm is allowed to query the value of the unknown target function  $f^*$  on a polynomial number of sets  $S_1, S_2, \dots$ , that may be chosen in an adaptive fashion. The algorithm must then output in polynomial time a function  $f$  that approximates  $f^*$  everywhere, namely  $f(S) \leq f^*(S) \leq \alpha f(S), \forall S \subseteq [n]$ . A formal definition of this model and the results are presented in Section 5.

**Learning with Prices.** This framework aims to model economic interactions more realistically and considers a setting where an agent with the target valuation  $f^*$  is interested in purchasing bundles of goods. In this framework, the learner does not obtain the value of  $f^*$  on each input set  $S_1, S_2, \dots$ . Instead, for each input set  $S_l$  the learner quotes a price  $p_l$  on  $S_l$  and observes 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  $\alpha$  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. This framework and the associated results are presented in Section 6.

### 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. In contrast, there is a  $\tilde{\Theta}(n^{1/6})$  gap between the existing bounds for submodular valuations [5].

<sup>5</sup>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.

Furthermore, we reveal the importance of considering the complexity of the target function (in a natural representation) for polynomial-time PMAC learning. We show that XOS valuations representable with a polynomial number of SUM trees are PMAC-learnable to a  $n^\eta$  factor in time  $n^{1/\eta}$ , for any  $\eta > 0$ . Finally, we show that XOS valuations representable with an arbitrary number of SUM trees, each with at most  $R$  leaves, are PMAC-learnable to an  $R$  factor.

### 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. We note that our lower bound construction is much simpler and gives a better bound than the  $\Omega(n^{1/3}/\log n)$  construction of [5]. However, the latter construction is for matroid rank functions, a significantly smaller class. For our upper bounds we provide structural results showing that XOS and subadditive functions can be approximated by a linear function to an  $O(\sqrt{n})$  and  $O(\sqrt{n} \ln n)$  factor respectively. We can then PMAC-learn these classes via a reduction to the classical problem of PAC-learning a linear separator.

**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 exist sets  $A_1, A_2, \dots, A_k \subseteq [n]$  such that
- (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 where we construct each  $A_i$  by picking each element in  $[n]$  with probability  $\frac{1}{\sqrt{n}}$ . 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,

$$\Pr[\sqrt{n}/2 < Y_i < 2\sqrt{n}] > 1 - 2e^{-\sqrt{n}/8} \quad \text{and} \quad \Pr[X_{ij} > \ln n] < \frac{e^{\ln n}}{\ln n^{\ln n}} = n^{-(\ln \ln n - 1)}, \quad \forall 1 \leq i < j \leq k.$$

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

Given the existence of the family  $\mathcal{A} = \{A_1, \dots, A_k\}$  of sets with properties (i) and (ii) above, 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  $\text{poly}(n, \frac{1}{\epsilon}, \frac{1}{\delta})$ . 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  $f : 2^{[n]} \rightarrow \mathbb{R}_+$  be a non-negative XOS function with  $f(\emptyset) = 0$ . Then there exists a function  $\hat{f}$  of the form  $\hat{f}(S) = \sqrt{w^T \chi(S)}$  where  $w \in \mathbb{R}_+^n$  such that  $\hat{f}(S) \leq f(S) \leq \sqrt{n} \hat{f}(S)$  for all  $S \subseteq [n]$ .*

*Proof.* XOS valuations are known [15] to be equivalent to fractionally subadditive valuations. A function  $f : 2^{[n]} \rightarrow \mathbb{R}$  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  $s \in T$ .

We can show the following property of XOS valuations: for any XOS  $f$  we have  $f(T) = \max\{\sum_{i \in T} x_i \mid x \in P(f)\}$ , where  $P(f)$  is the associated polyhedron  $\{x \in \mathbb{R}_+^n : \sum_{i \in S} x_i \leq f(S), \forall S \subseteq [n]\}$ . 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  $f(T)$ . Formally, for any  $T \subseteq [n]$  we have  $\sum_{i \in T} x_i \leq f(T)$  for any  $x \in P(f)$ . Therefore  $f(T) \geq \max \{ \sum_{i \in T} x_i \mid x \in P(f) \}$ . Now we prove that in fact

$$f(T) \leq \max \left\{ \sum_{i \in T} x_i \mid x \in P(f) \right\}.$$

Consider the linear programming (LP1) for the quantity  $\max \{x(T) \mid x \in P(f)\}$  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.

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

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

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

Since  $f$  is fractionally subadditive and  $\sum_{S: i \in S} y_S^* \geq 1, \forall i \in T$ , we have  $f(T) \leq \sum_{S \subseteq [n]} y_S^* f(S)$ , hence  $f(T) \leq \max \{x(T) \mid x \in P(f)\}$ . This completes the proof of the fact that  $f(T) = \max \{ \sum_{i \in T} x_i \mid x \in P(f) \}$ .

Given this result, we proceed as follows (a very similar approach is used by [17] for submodular functions). Define  $P = \{x \in \mathbb{R}^n : (|x_1|, \dots, |x_n|) \in P(f)\}$ . Since  $P$  is bounded and *central symmetric* (i.e.  $x \in P \Leftrightarrow -x \in P$ ), there exists [20] 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  $\hat{f}(T) \leq f(T) \leq \sqrt{n}\hat{f}(T), \forall T \subseteq [n]$ . At last, basic calculus implies  $\hat{f}(T) = \sqrt{w^\top \chi(T)}$  for some  $w \in \mathbb{R}_+^n$ .  $\square$

For PMAC-learning XOS valuations to with an approximation factor of  $\sqrt{n + \epsilon}$ , we apply Algorithm 1 with parameters  $R = n$ ,  $\epsilon$ , and  $p = 2$ . The proof of correctness of Algorithm 1 follows by using the structural result in Claim 1 and a technique of [5] that we sketch briefly here. Full details of this proof appear in Appendix A.

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  $nw^\top \chi(S) - (f^*(S))^2 \geq 0$  and  $nw^\top \chi(S) - (n + \epsilon)(f^*(S))^2 < 0$ .

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

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 [21, 33]. 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  $\text{ex}_S^+$  i.e.  $+1$  if the coin is heads and  $\text{ex}_S^-$  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  $u^\top x = 0$ , where  $u = (\hat{w}, -z)$ ,  $w \in \mathbb{R}^n$  and  $z > 0$ . The key observation is that the function  $f(S) = \frac{1}{(n + \epsilon)z} \hat{w}^\top \chi(S)$  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  $\epsilon$  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 note that any subadditive valuation can be approximated by an XOS valuation to a  $\ln n$  factor [12, 6]<sup>6</sup> 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  $R = n \ln^2 n$ ,  $\epsilon$ , and  $p = 2$ . Correctness then follows by a reasoning similar to the one for XOS functions.  $\square$

---

**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  $\mathcal{S}_{\neq 0} = \{(A_i, f^*(A_i)) \in \mathcal{S} : f^*(A_i) \neq 0\} \subseteq \mathcal{S}$  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  $x_i = \begin{cases} (\chi(A_i), (f^*(A_i))^p) & \text{(if } y_i = +1) \\ (\chi(A_i), (R + \epsilon) \cdot (f^*(A_i))^p) & \text{(if } y_i = -1). \end{cases}$

- 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  $(x, \text{sgn}(u^\top x))$  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  $f(S) = \left( \frac{1}{(R+\epsilon)z} \hat{w}^\top \chi(S) \right)^{1/p}$ .

---

### 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. 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 not care about computational complexity. Interestingly we can show that we can achieve good PMAC learnability via polynomial time algorithms. In particular, we show that XOS functions representable with at most  $R$  SUM trees can be PMAC-learned with a  $R^\eta$  approximation factor in time  $n^{O(1/\eta)}$ , for any  $\eta > 0$ . 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  $n^\eta$ , in time  $n^{O(1/\eta)}$ , for any  $\eta > 0$ .

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

*Proof.* Let  $L = 1/\eta$  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

$$f'(S) \leq (f^*(S))^L \leq R f'(S), \quad \forall S \subseteq [n] \quad \text{i.e.} \quad 1/R \sum_j k_j^L(S) \leq \max_j k_j^L(S) \leq \sum_j k_j^L(S), \quad \forall S \subseteq [n] \quad (1)$$

The left-hand side inequalities in Eq. (1) 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. (1) 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

---

<sup>6</sup>We are grateful to Shahar Dobzinski and Kshipra Bhawalkar for pointing out this fact to us.

over this new set of features.  $\chi_M(S)_{i_1, i_2, \dots, i_L} = 1$  if all items  $i_1, i_2, \dots, i_L$  appear in  $S$  and  $\chi_M(S)_{i_1, i_2, \dots, i_L} = 0$  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  $R, \epsilon$ , 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  $(R+\epsilon)^{1/L}$  factor on any point  $\chi_M(S)$  corresponding to sets  $S \subseteq [n]$  from a collection  $\mathcal{S}$  with at least an  $1-\epsilon$  measure in  $D$ , i.e.  $f''(\chi_M(S)) \leq f^*(S) \leq (R+\epsilon)^{1/L} f''(\chi_M(S))$ . 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. for any  $S \in \mathcal{S}$  we have

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

As desired, with high confidence the hypothesis  $f$  approximates  $f^*$  to a  $(R+\epsilon)^n$  factor on most sets from  $D$ .  $\square$

This result has an appealing interpretation in terms of representations of submodular functions. We know that any submodular function is representable as an XOS tree. What Theorem 2 implies is that (submodular) functions that are succinctly representable as XOS trees can be PMAC-learned well. Theorem 2 is thus a target-dependent learnability result, in that the extent of learnability of a function depends on the function's complexity.

### 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. 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. We now show good PMAC-learning guarantees for classes of functions of this type.

**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  $m = O(\frac{1}{\epsilon} (n \log \log_{1+\eta}(\frac{H}{h}) + \log(1/\delta)))$  and running time polynomial in  $m$ , where  $h$  and  $H$  are the smallest and the largest non-zero values our functions can take.*

*Proof.* We show that the unit-demand hypothesis  $f$  output by Algorithm 2 produces the desired result. The algorithm constructs 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 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:

$$f(S_l) \leq f^*(S_l) \leq Rf(S_l) \quad \text{for all } l \in \{1, 2, \dots, m\}. \quad (2)$$

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

$$f(i) \geq f^*(i) \quad \text{for any } i \in S_1 \cup \dots \cup S_m. \quad (3)$$

Therefore for any  $l \in \{1, 2, \dots, m\}$ . :

$$f^*(S_l) \leq R \max_{i \in S_l} f^*(i) \leq R \max_{i \in S_l} f(i) = Rf(S_l),$$

where the first inequality follows by definition, and the second inequality follow from relation (3). 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 (2), as desired.

---

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

---

**Input:** A sequence of training examples  $\mathcal{S} = \{(S_1, f^*(S_1)), (S_2, f^*(S_2)), \dots, (S_m, f^*(S_m))\}$ .

- 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  $S \subseteq \{1, \dots, n\}$ .

---

To finish the proof we show that  $m = O(\frac{1}{\epsilon} (n \log \log_{1+\eta}(\frac{H}{h}) + \log(1/\delta)))$  is sufficient so that with probability at least  $1 - \delta$   $f$  approximates the target function  $f^*$  multiplicatively within a factor of  $R(1 + \eta)^2$  on a  $1 - \epsilon$  fraction of the distribution. Let  $F_\eta$  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  $m = O(\frac{1}{\epsilon} (n \log \log_{1+\eta}(\frac{H}{h}) + \log(1/\delta)))$  examples are sufficient such that any function in  $F_\eta$  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  $F_\eta$  is a multiplicative  $L_\infty$  cover for the class of unit-demand functions, we easily get the desired result [1].  $\square$

## 4 PMAC-learnability of OXS and Gross Substitutes Valuations

In this section we study the learnability of subclasses of submodular valuations, namely OXS and gross substitutes. We start by focusing on interesting subclasses of OXS functions that arise in practice, namely OXS functions representable with a small number of MAX trees or leaves<sup>7</sup>. 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:

**Theorem 4.** (1) Let  $\mathcal{F}$  be the family of OXS functions representable with at most  $R$  MAX trees. For any  $\eta$ , the family  $\mathcal{F}$  is properly PMAC-learnable with approximation factor of  $R(1+\eta)$  by using  $m = O(\frac{1}{\epsilon} (n \log \log_{1+\eta}(\frac{H}{h}) + \log(1/\delta)))$  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  $O(n^R \log(n/\delta)/\epsilon)$  training examples and running time  $\text{poly}(n, 1/\epsilon, 1/\delta)$ .

(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  $O(\frac{n}{\epsilon} \log(\frac{n}{\delta\epsilon}))$  training examples and running time  $\text{poly}(n, 1/\epsilon, 1/\delta)$ .

*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 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 fact that  $\mathcal{F}$  is learnable to a factor of  $R(1 + \eta)$  for any  $\eta$  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  $m = O(n \ln(n/\delta)/\epsilon)$  training examples and time  $\text{poly}(n, 1/\epsilon, 1/\delta)$  – see Lemma 3 in Appendix B. Second, it is easy

---

<sup>7</sup> We note that the literature on algorithms for secretary problems [3, 4] often considers a subclass of the latter class, in which each item must have the same value in any tree.

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 4 in Appendix B). These two facts together imply that for constant  $R$ , the class  $\mathcal{F}$  is PAC-learnable by using  $O(n^R \log(n/\delta)/\epsilon)$  training examples and running time  $\text{poly}(n, 1/\epsilon, 1/\delta)$ .

(2) We start by showing the following structural result: if  $f^*$  has an OXSrepresentation 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  $S \subseteq \{1 \dots n\}$  satisfies

$$f^*(S) \leq f(S) \leq R \cdot f^*(S) \quad \text{for all } S \subseteq \{1 \dots n\} \quad (4)$$

By subadditivity,  $f^*(S) \leq Rf(S)$ , for all  $S$ . Let  $f^*_{1}, \dots, f^*_{k}$  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  $(S_1, \dots, S_k)$  of  $S$  in which item  $i$  is mapped to  $S_{j_i}$  for any  $i$ , we have  $\sum_{i \in S} f^*(i) \leq Rf^*_{1}(S_1) + \dots + Rf^*_{k}(S_k)$ . Therefore:

$$f(S) = \frac{1}{R} \sum_{i \in S} f^*(i) \leq \max_{(S_1, \dots, S_k) \text{ partition of } S} (f^*_{1}(S_1) + \dots + f^*_{k}(S_k)) = f^*(S),$$

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

Given the structural result 4, we can PMAC-learn the class of OXS functions representable with MAX trees with at most  $R$  leaves  $y$  using Algorithm 1 with parameters  $R, \epsilon$  and  $p = 1$ . The correctness by using a reasoning similar to the one in Theorem 1.  $\square$

We now consider the class of Gross Substitutes valuations, a superclass of OXS valuations and a subclass of submodular valuations (recall Lemma 1). Gross Substitutes are fundamental to allocation problems with per-item prices [10, 18, 28]; in particular a set of per-item market-clearing prices exists if and (almost) only if all customers have gross substitutes valuations. A valuation is gross substitutes if raising prices on some items preserves the demand on other items. Given prices on items, an agent with valuation  $f$  demands a preferred set, formalized as follows.

**Definition 5.** For price vector  $\vec{p} \in \mathbb{R}^n$ , the demand correspondence  $\mathcal{D}_f(\vec{p})$  of valuation  $f$  is the collection of preferred sets at prices  $\vec{p}$ , i.e.  $\mathcal{D}_f(\vec{p}) = \arg \max_{S \subseteq \{1, \dots, n\}} \{f(S) - \sum_{j \in S} p_j\}$ . A valuation  $f$  is gross substitutes (GS) if for any price vectors  $\vec{p}' \geq \vec{p}$  (i.e.  $p'_i \geq p_i \forall i \in [n]$ ), and any  $A \in \mathcal{D}_f(\vec{p})$  there exists  $A' \in \mathcal{D}_f(\vec{p}')$  with  $A' \supseteq \{i \in A : p_i = p'_i\}$ .

That is, the GS property requires that all items  $i$  in some preferred set  $A$  at the old prices  $\vec{p}$  and for which the old and new prices are equal ( $p_i = p'_i$ ) are simultaneously contained in some preferred set  $A'$  at the new prices  $\vec{p}'$ .

As mentioned earlier Balcan and Harvey [5] proved that it is hard to PMAC-learn the class of submodular functions with an approximation factor  $o(n^{1/3}/\log n)$ . We show here that their result applies even for the class of gross substitutes. This is quite surprising since such functions are typically considered easy from an economic optimization point of view. Specifically:

**Theorem 5.** No algorithm can PMAC-learn the class of gross substitutes with an approximation factor of  $o(n^{1/3}/\log n)$ . This holds even if  $D$  is known and value queries are allowed.

*Proof.* It is known that the class of matroid rank functions cannot be PMAC-learned with an approximation factor of  $o(n^{1/3}/\log n)$ , even if  $D$  is known and value queries are allowed [5]. One can show that a matroid rank function is a gross substitutes function (see Lemma 2 below). Combining these, yields the theorem.  $\square$

Our key tool for proving that any matroid rank function is also GS (Lemma 2 below) is a valuation-based characterization of gross substitutes valuations due to [25].

**Lemma 2.** A matroid rank function is gross substitutes.

*Proof.* Denote  $f$ 's marginal value over  $S$  by  $f^S(A) = f(S \cup A) - f(S), \forall A \subseteq [n] \setminus S$ . As shown in [25]  $f$  is GS if and only if

$$f^S(ab) + f^S(c) \leq \max\{f^S(ac) + f^S(b), f^S(bc) + f^S(a)\} \quad \text{for all items } a, b, c \text{ and set } S \quad (5)$$

i.e. (by taking permutations) there is no unique maximizer among  $f^S(ab) + f^S(c), f^S(ac) + f^S(b), f^S(bc) + f^S(a)$ .

If  $f$  is matroid rank function, then so is  $f^S$ ; in particular,  $f^S(A) \leq |A|, \forall A \subseteq [n] \setminus S$ . We reason by case analysis.

Suppose that  $f^S(ab) = 2$ . Then we have  $f^S(a) = f^S(b) = 1$ . If  $f^S(ac) = 2$  or  $f^S(bc) = 2$ , then  $f^S(c) = 1$  and hence the inequality (5) holds. On the other hand, if  $f^S(ac) = f^S(bc) = 1$ , then we have by the monotonicity and submodularity of  $f^S$ ,  $f^S(ab) + f^S(c) \leq f^S(abc) + f^S(c) \leq f^S(ac) + f^S(bc) = 2$ , and the inequality (5) holds.

If  $f^S(ab) \leq 1$  then  $f^S(ab) = \max\{f^S(a), f^S(b)\}$ . As  $f^S(c) \leq f^S(ac)$  and  $f^S(c) \leq f^S(bc)$ , Eq. (5) follows.  $\square$

We note that Lemma 2 was previously proven in [26] in a more involved way via the concept of  $M^{\natural}$ -concavity from discrete convex analysis.

## 5 Learnability everywhere with value queries

In this section, we consider approximate learning with value queries [17, 31]. 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 6.** *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  $f(S) \leq f^*(S) \leq \alpha f(S), \forall S \subseteq \{1, \dots, n\}$ .*

Goemans et al. [17] 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, efficiently achieving a factor of  $R$ .

**Theorem 6. (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 Sketch. (1)* We show in Appendix C that the family of valuation functions used in [17] 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  $g^R(S) = \min(\beta + |S \cap ((\{1, \dots, n\} \setminus R)|, |S|, \alpha')$  for  $\alpha' = xn^{1/2}/5, \beta = x^2/5$  with  $x^2 = \omega(\log n)$  and  $R$  a subset of  $\{1, \dots, n\}$  of size  $\alpha'$  (chosen uniformly at random). These valuations are OXS; for example,  $g_{23}(S)$  can be expressed as a SUM of  $\alpha'$  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  $f$ .  $\square$

**Note:** We note that the lower bound technique in [17] has been later used in a sequence of papers [19, 16, 31] concerning optimization under submodular cost functions and our result (Lemma 6 in particular) implies that all the lower bounds in these papers apply to the smaller classes of OXS functions and GS functions.

**Note:** We also note that since XOS contains all submodular valuations, the lower bound of Goemans et al. [17] 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 [17].

## 6 Learning with prices

We now introduce 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  $S_1, S_2, \dots$ . 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 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  $\alpha$  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 [28], where prices are:  $p_l$  on  $S_l$  and  $\infty$  elsewhere. We call PMAC-learning with prices and VQ-learning with prices the variants of this framework applied to our two learning 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  $\beta$  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 7 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 (standard) PMAC learning. For convenience, we assume in this section 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 7.** *Consider a family  $\mathcal{F}$  of valuations such that the  $p$ -th power of any  $f^* \in \mathcal{F}$  can be approximated to a  $\beta$  factor by a linear function: i.e., for some  $w$  we have  $w^\top \chi(S) \leq (f^*(S))^p \leq \beta w^\top \chi(S)$  for all  $S \subseteq [n]$ , where  $\beta \geq 1, p > 0$ . Then for any  $0 < \eta \leq 1$ , the family  $\mathcal{F}$  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  $\text{poly}(n, \frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{\eta})$ .*

*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  $w^\top \chi(S_l)$ , 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 yields 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  $i = 0, 1, 2, \dots, N + 1$ , 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  $\text{sgn}((\beta w, -1)^\top x_l)$  in each case:  $\beta w^\top \chi(S_l) \leq \beta (f^*(S_l))^p < \beta q_l^p$  and  $q_l^p \leq (f^*(S_l))^p \leq \beta w^\top \chi(S_l)$  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  $f'(S) = \frac{1}{\beta z} \hat{w}^\top \chi(S)$  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  $\varepsilon$  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  $\hat{w}^\top \chi(S) > \beta z [(1 + \eta/3)f^*(S)]^p$ . 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  $\hat{w}^\top \chi(S) > \beta z q^p$ . Thus the separator mistakenly predicts positive.
2. Alternatively it could be that  $(1 + \eta/3)^2 \beta^{1/p} f(S) < f^*(S)$ . This implies that  $(1 + \eta/3) \beta^{1/p} f'(S)^{1/p} < f^*(S)$  or equivalently that  $\hat{w}^\top \chi(S) < z (\frac{f^*(S)}{1 + \eta/3})^p$ . 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  $\hat{w}^\top \chi(S) < z q^p$ . 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.  $\square$

**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  $1 - \tilde{\epsilon}$  and a uniformly at random price in  $\{1, 2, 4, \dots, H/2, H\}$  as in the previous result with probability  $\tilde{\epsilon}$ . The sample complexity of learning only goes up by a factor of at most  $\frac{\log H \log \log H}{\tilde{\epsilon}}$ .

We can also recover our upper bounds on learnability everywhere with value queries (the corresponding lower bounds clearly hold). By sequentially setting prices  $1, 2, 4, \dots, H/2, H$  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  $f^*(\{1\}), \dots, f^*(\{n\})$  then yield the VQ-learnability with prices of these classes.

**Theorem 8.** *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.*

## 7 Conclusions

In this paper we study the approximate learnability of valuations commonly used throughout economics and game theory for the quantitative encoding of agent preferences. We provide upper and lower bounds regarding the learnability of important subclasses of valuation functions that express no-complementarities. Our main results concern their approximate learnability in the distributional learning (PAC-style) setting. We provide nearly tight lower and upper bounds of  $\tilde{\Theta}(n^{1/2})$  on the approximation factor for learning XOS and subadditive valuations, both widely studied superclasses of submodular valuations. Interestingly, we show that the  $\tilde{\Omega}(n^{1/2})$  lower bound can be circumvented for XOS functions of polynomial complexity; we provide an algorithm for learning the class of XOS valuations with a representation of polynomial size achieving an  $O(n^\epsilon)$  approximation factor in time  $O(n^{1/\epsilon})$  for any  $\epsilon > 0$ . This highlights the importance of considering the complexity of the target function for polynomial time learning. We also provide new learning results for interesting subclasses of submodular functions. Our upper bounds for distributional learning leverage novel structural results for all these valuation classes. We show that many of these results provide new learnability results in the Goemans et al. model [17] of approximate learning everywhere via value queries.

We also introduce a new model that is more realistic in economic settings, in which the learner can set prices and observe purchase decisions at these prices rather than observing the valuation function directly. In this model, most of our upper bounds continue to hold despite the fact that the learner receives less information (both for learning in the distributional setting and with value queries), while our lower bounds naturally extend.

**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, and a Microsoft Research Faculty Fellowship.

## References

- [1] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.
- [2] L. M. Ausubel and P. R. Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1, 2002.
- [3] M. Babaioff, M. Dinitz, A. Gupta, N. Immorlica, and K. Talwar. Secretary problems: weights and discounts. In *SODA*, 2009.
- [4] 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.
- [5] M. F. Balcan and N. Harvey. Learning submodular functions. In *STOC*, 2011.
- [6] K. Bhawalkar and T. Roughgarden. Welfare guarantees for combinatorial auctions with item bidding. In *SODA*, 2011.
- [7] A. Blum, M. Zinkevich, and T. Sandholm. On polynomial-time preference elicitation with value queries. In *ACM Conference on Electronic Commerce*, 2003.
- [8] 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 *SODA*, pages 518–536, 2010.
- [9] 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, 2010.
- [10] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.
- [11] R. Day and S. Raghavan. Assignment preferences and combinatorial auctions. Working paper, University of Connecticut, April 2006.
- [12] S. Dobzinski. Two randomized mechanisms for combinatorial auctions. In *APPROX*, 2007.
- [13] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. *STOC '05*, pages 610–618, 2005.
- [14] S. Dobzinski, N. Nisan, and M. Schapira. Truthful randomized mechanisms for combinatorial auctions. *STOC '06*, pages 644–652, 2006.
- [15] 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.
- [16] 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.
- [17] M. Goemans, N. Harvey, S. Iwata, and V. Mirrokni. Approximating submodular functions everywhere. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2009.
- [18] F. Gul and E. Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87(1):95–124, 1999.
- [19] S. Iwata and K. Nagano. Submodular function minimization under covering constraints. In *Proceedings of the 50th Annual Symposium on Foundations of Computer Science*, 2009.
- [20] 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.

- [21] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.
- [22] S. Lahaie, F. Constantin, and D. C. 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.
- [23] S. Lahaie and D. C. Parkes. Applying learning algorithms to preference elicitation. In *ACM Conference on Electronic Commerce*, pages 180–188, 2004.
- [24] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *ACM Conference on Electronic Commerce*, pages 18–28, 2001.
- [25] Y. Lien and J. Yan. On the gross substitutes condition. Working paper, Jul2007.
- [26] K. Murota. Submodular function minimization and maximization in discrete convex analysis. Technical Report METR 2008–32, The University of Tokyo, Japan, August 2008.
- [27] N. Nisan. Chapter 9: Bidding Languages for Combinatorial Auctions . In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*. MIT Press, 2006.
- [28] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge, 2007.
- [29] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, pages 542–547, 2001.
- [30] Y. Singer. Budget feasible mechanisms. FOCS ’10, pages 765–774, 2010.
- [31] Z. Svitkina and L. Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008.
- [32] L. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.
- [33] V. N. Vapnik. *Statistical Learning Theory*. Wiley and Sons, 1998.

## A 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  $R = n$ ,  $\epsilon$ , and  $p = 2$ . The proof of correctness of Algorithm 1 follows by using the structural result in Claim 1 and a technique of [5]. We provide here the full details of this proof.

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:

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

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 [21, 33] 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  $c^* : \mathbb{R}^m \rightarrow \{-1, +1\}$ ,  $c^*(x) = \text{sgn}(u^\top x)$ . The examples induced by  $D'$  and  $c^*$  are called *linearly separable* since there exists a vector  $u$  such that  $c^*(x) = \text{sgn}(u^\top x)$ . 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

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

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  $f(S) = \hat{w}^\top \chi(S)$  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

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

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,

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

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

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

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

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

Using  $\mathcal{S}_{\neq 0}$ , the algorithm then constructs a sequence  $\mathcal{S}'_{\neq 0} = ((x_1, y_1), \dots, (x_a, y_a))$  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  $f(S) = (\frac{1}{(n+\varepsilon)z} \hat{w}^\top \chi(S))^{1/2}$ .

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  $\text{sgn}(u^\top x)$ , the linear separator found by our algorithm. It is easy to show that the function  $f(S) = (\frac{1}{(n+\varepsilon)z} \hat{w}^\top \chi(S))^{1/2}$  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

$$\begin{aligned} \hat{w}^\top \chi(S) - z(f^*(S))^2 > 0 \quad \text{and} \quad \hat{w}^\top \chi(S) - z(n + \varepsilon)(f^*(S))^2 < 0 \\ \implies \frac{1}{(n + \varepsilon)z} \hat{w}^\top \chi(S) < (f^*(S))^2 < (n + \varepsilon) \frac{1}{(n + \varepsilon)z} \hat{w}^\top \chi(S). \end{aligned}$$

So, for any point in  $S \in \mathcal{Y}$ , the function  $f(S)^2 = \frac{1}{(n+\varepsilon)z} \hat{w}^\top \chi(S)$  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  $m = \frac{16n}{\varepsilon} \log(\frac{n}{\delta\varepsilon})$ , 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  $\varepsilon$ .*

*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  $\varepsilon$ . Define  $k' = k + \log(n/\delta)/\varepsilon$ . Then amongst the subsequent examples  $S_{k+1}, \dots, S_{k'}$ , the probability that none of them lie in  $\mathcal{Z} \setminus \mathcal{P}_k$  is at most  $(1 - \varepsilon)^{\log(n/\delta)/\varepsilon} \leq \delta/n$ . 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  $m \geq n \log(n/\delta)/\varepsilon$ , with probability at least  $\delta$  the set  $\mathcal{Z} \setminus \mathcal{P}_m$  has measure at most  $\varepsilon$ .  $\square$

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  $2\epsilon$  under  $D$ .

*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  $\epsilon$ . 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  $a \geq 8n \log(n/\delta\epsilon) \frac{q}{\epsilon}$ . By a standard sample complexity argument [33] 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  $\epsilon/q$  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  $(\chi(S), (n + \epsilon) \cdot (f^*(S))^2)$  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.  $\square$

In summary, our algorithm outputs a hypothesis  $f$  approximating  $f^*$  to within a factor  $(n + \epsilon)^{1/2}$  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  $3\epsilon$ , with probability at least  $1 - 3\delta$ .

## B 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 3.** *The class of unit-demand valuations is properly PAC-learnable by using  $m = O(n \ln(n/\delta)/\epsilon)$  training examples and time  $\text{poly}(n, 1/\epsilon, 1/\delta)$ .*

*Proof.* We first show how to solve the consistency problem in polynomial time: given a sample  $(S_1, f^*(S_1)), \dots, (S_m, f^*(S_m))$  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  $l \in \{1, 2, \dots, m\}$ . 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  $l \in \{1, 2, \dots, m\}$ . So  $f(i) \geq f^*(i)$  for any  $i \in S_1 \cup \dots \cup S_m$ . Therefore for any  $l \in \{1, 2, \dots, m\}$  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  $l \in \{1, 2, \dots, m\}$ .

We now claim that  $m = O(n \ln(n/\delta)/\epsilon)$  training examples are sufficient so that with probability at least  $1 - \delta$ , the hypothesis  $f$  produced has error at most  $\epsilon$ . In particular, notice that Algorithm 2 guarantees that  $f(i) \in \{f^*(1), \dots, f^*(n)\}$  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  $\epsilon$  is at most  $n^n(1 - \epsilon)^m$  which is at most  $\delta$  for our given choice of  $m$ .  $\square$

**Lemma 4.** *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  $O(Rn^R \log(n/\delta)/\epsilon)$  training examples and time  $\text{poly}(n, 1/\epsilon, 1/\delta)$ .*

*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,

$$f^*(S) = \max_{\substack{S^R \subseteq S \\ |S^R| \leq R}} f^*(S^R), \forall S \subseteq \{1, \dots, n\} \quad (6)$$

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  $S^R \subseteq \{1, \dots, n\}$  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

$$f'(\{i_{S_1^R}, \dots, i_{S_L^R}\}) = \max_{l=1, \dots, L} f'(i_{S_l^R}). \quad (7)$$

By equations (6) and (7), 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 3 guarantees that this will PAC-learn using  $O(n^R \log(n^R/\delta)/\epsilon)$  training examples and running time  $\text{poly}(n, 1/\epsilon, 1/\delta)$  for constant  $R$ .  $\square$

## C Additional Result for Theorem 6

[17] 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).  $f_{R, \alpha', \beta}(\cdot) : 2^{\{1, \dots, n\}} \rightarrow \mathbb{R}$  is defined as follows. Let subset  $R \subseteq \{1, \dots, n\}$  and  $\bar{R} = (\{1, \dots, n\}) \setminus R$  its complement. Also fix integers  $\alpha', \beta \in \mathbb{N}$ . Then

$$f_{R, \alpha', \beta}(S) = \min(\beta + |S \cap \bar{R}|, |S|, \alpha'), \forall S \subseteq \{1, \dots, n\} \quad (8)$$

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

$$f_{R', c}(S) = \min(c, |S \cap R'|), \forall S \subseteq \{1, \dots, n\} \quad (9)$$

*is in OXS.*

*Proof.* For ease of notation let  $f(\cdot) = f_{R', c}(\cdot)$ . We assume  $c < |R'|$ ; otherwise,  $f(S) = |S \cap R'|, \forall S \subseteq \{1, \dots, n\}$ , 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 [24]. 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(a).

Then  $T(S) = f(S), \forall S \subseteq \{1, \dots, n\}$  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  $T(S) \leq c, \forall S \subseteq \{1, \dots, n\}$ .  $\square$

**Lemma 6.** *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<sup>8</sup>  $|S| \leq \alpha', \forall S \subseteq \{1, \dots, n\}$  and

$$f(S) = \min(\beta + |S \cap \bar{R}|, |S|) = |S \cap \bar{R}| + \min(\beta, |S \cap R|) \quad (10)$$

<sup>8</sup>We note that  $n > \alpha'$  in [17]. We consider this case for completeness.

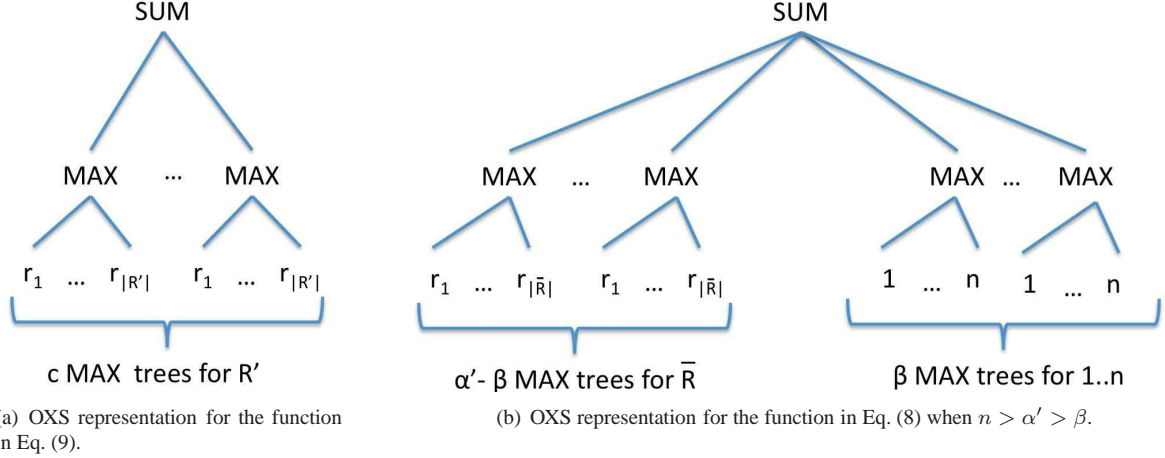


Figure 1: OXS representations. All leaves have value 1.

From the proof of Lemma 5 we get that the function  $f'(S) = \min(\beta, |S \cap R|)$  has an OXS tree  $T'$  (i.e.  $T'(S) = f'(S), \forall S \subseteq \{1, \dots, n\}$ ) with  $\beta$  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  $T(S) = f(S), \forall S \subseteq \{1, \dots, n\}$ . The additional  $|\bar{R}|$  MAX trees encode the  $|S \cap \bar{R}|$  term in Eq. (10). If  $\alpha' \leq \beta$  then  $f(S) = \min(\alpha', |S|)$ ; the claim follows by Lemma 5 for  $c = \alpha', R' = \{1, \dots, n\}$ . 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(b).

- $\alpha' - \beta$  MAX trees  $T_1 \dots T_{\alpha' - \beta}$ , each having as leaves all the elements in  $\bar{R}$  with value 1.
- $\beta$  MAX trees  $T_{\alpha' - \beta + 1} \dots T_{\alpha'}$ , each having as leaves all the elements (in  $\{1, \dots, n\}$ ) 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  $\alpha'$  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  $T_{\alpha' - \beta + 1} \dots T_{\alpha'}$ .
- $f(S) = \alpha' \leq \min(\beta + |S \cap \bar{R}|, |S|)$ . 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  $T_1 \dots T_{\alpha' - \beta}$ . Also  $|S| \geq \alpha'$  and we can take the remaining  $\beta$  elements from  $T_{\alpha' - \beta + 1} \dots T_{\alpha'}$ .
- $f(S) = |S| \leq \min(\beta + |S \cap \bar{R}|, \alpha')$ . 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  $T_1 \dots T_{\alpha' - \beta}$  (and from  $T_{\alpha' - \beta + 1} \dots T_{\alpha'}$  if  $|S \cap \bar{R}| > \alpha' - \beta$ ) and elements in  $S \cap R$  from  $T_{\alpha' - \beta + 1} \dots T_{\alpha'}$ .
- $f(S) = \beta + |S \cap \bar{R}| \leq \min(|S|, \alpha')$ . We claim  $T(S) \geq \beta + |S \cap \bar{R}|$ : we can take  $\beta \leq |S \cap R|$  elements from  $T_{\alpha' - \beta + 1} \dots T_{\alpha'}$  and  $|S \cap \bar{R}| \leq \alpha' - \beta$  elements from  $T_1 \dots T_{\alpha' - \beta}$ . Finally,  $T(S) \leq \beta + |S \cap \bar{R}|$  since at most all elements in  $S \cap \bar{R}$  can be taken from  $T_1 \dots T_{\alpha' - \beta}$  and at most  $\beta$  elements in  $S \cap R$  from  $T_{\alpha' - \beta + 1} \dots T_{\alpha'}$ .

□