Sampling and Representation Complexity of Revenue Maximization

Shaddin Dughmi[1]⋆, Li Han[1]⋆⋆, and Noam Nisan[2]⋆⋆⋆

[1] University of Southern California
{shaddin,han554}@usc.edu
[2] Hebrew University and Microsoft Research
noam@cs.huji.ac.il

Abstract. We consider (approximate) revenue maximization in mechanisms where the distribution on input valuations is given via "black box" access to samples from the distribution. We analyze the following model: a single agent, m outcomes, and valuations represented as m-dimensional vectors indexed by the outcomes and drawn from an arbitrary distribution presented as a black box. We observe that the number of samples required – the sample complexity – is tightly related to the representation complexity of an approximately revenue-maximizing auction. Our main results are upper bounds and an exponential lower bound on these complexities. We also observe that the computational task of "learning" a good mechanism from a sample is nontrivial, requiring careful use of regularization in order to avoid over-fitting the mechanism to the sample. We establish preliminary positive and negative results pertaining to the computational complexity of learning a good mechanism for the original distribution by operating on a sample from said distribution.

1 Introduction

In the general (quasi-linear, independent-private-value, Bayesian) mechanism design setting, a principal must choose from one from a set A of *outcomes*, and there are n *bidders* each of whom has a *valuation* $v_i : A \to \Re$ mapping outcomes to real values. These valuations are private, and the mechanism designer only knows that each v_i is drawn from a *prior distribution* $\mathcal{D}_i$ on valuations. Based on these distributions, the mechanism designer must design a *mechanism* that determines, for each profile of bidder valuations, an outcome which may be probabilistic – a *lottery* – and a *payment* from each player. The rational behavior of the bidders is captured by two sets of constraints: *Incentive constraints* require that no bidder can improve his expected utility by behaving according to – i.e. "reporting to the mechanism" – another valuation v'_i. *Individual Rationality constraints* require that bidders never lose from participating in the mechanism. Under these two sets of constraints the mechanism designer's goal is to maximize his expected *revenue*.

⋆ Supported in part by NSF CAREER Award CCF-1350900.
⋆⋆ Supported by NSF grant CCF-1350900.
⋆⋆⋆ For a full version see "L. Han, S. Dughmi, N. Nisan, Sampling and Representation Complexity of Revenue Maximization, arXiv:1402.4535, 2014".

Myerson's classical work [18] completely solves the problem for the special case of *single parameter* valuations, where each v_i is effectively captured by a scalar. In this case, the optimal mechanism has a very simple form and is deterministic. It turns out that this completely breaks down once we leave the single-parameter settings and it is known that deterministic mechanisms can be significantly inferior to general ones that are allowed to allocate lotteries [4, 11, 16, 17, 19], and that good revenue may require complex mechanisms [14]. Moreover, this is so even in very simple settings such as auctions with a single bidder and two items.

It is well known that even in the general multi-parameter case, the revenue maximizing mechanism is obtained using linear optimization [4]. While this may seem encouraging for both characterization and computation of the optimal mechanism, there is a rub: this LP formulation hides several exponential blowups in the natural parameters in most settings. Two types of such blowups, and how to overcome them, have received considerable attention recently: an exponential blowup in n, the number of bidders, that is a result of the fact that we have variables for each *profile* of valuations (e.g. [1,2,6,8]), and the fact that, in the case of various multi-item auctions, m, the size of the outcome space, is naturally exponential in the number of items (e.g. [7, 10, 12]).

In this paper we study a third type of exponential blowup who's consequences are not yet fully understood: the size of the support of each $\mathcal{D}_i$ – the size of the valuation space – is naturally exponential in the number of outcomes m. Formally, it is often a continuum since a valuation assigns a real value to each alternative, but even once we discretize (as we certainly will have to do for any computational purpose), the valuation space is still exponential in m. Does this exponential blowup necessarily translate to the computational task, or can revenue be maximized in time polynomial in m?

For most bite of our main result, which is negative, we focus on the simplest scenario of this type, one that exhibits only this exponential blowup in the size of the valuation space, and no others. Specifically, we have a *single bidder* bidding for m abstract alternatives in A. For a single bidder, this setting is essentially equivalent[3] to an auction setting studied e.g. in [4, 11] in which A is a set of items for sale, a "unit demand" bidder is interested in acquiring at most a single item and has a potentially different value $v(j)$ for each item $j \in A$. Furthermore, for a single bidder, an auction is just a pricing scheme, giving a *menu* that assigns a price for each possible lottery $(x_1 ... x_m)$ where $x_j \geq 0$ is the probability of getting item j and $\sum_{j \in A} x_j \leq 1$.

As observed in [4], if $\mathcal{D}$ (the prior distribution over v) happens to have a "small" support then the linear program is small and we are done. However, this is typically not the case: even if we restrict all item values to be 0 or 1 there are 2^m possible valuations and the linear program is exponential. This raises the question of how $\mathcal{D}$ is represented. Sometimes $\mathcal{D}$ admits some special structure — e.g. it may be a product distribution over item values — permitting a succinct representation. In other cases it may come from "nature" and we will, in some sense, have to "learn" $\mathcal{D}$ in order to construct our mechanism. Both of these scenarios can be captured by a black box *sampling* model in which our access to $\mathcal{D}$ is by means of a sequence of samples v, each

[3] The formal distinction is that in the unit-demand setting, there is special outcome $*$ with fixed value $v(*) = 0$, denoting not allocating any item.

drawn independently at random from $\mathcal{D}$. Can we design an (approximately) revenue-maximizing auction for $\mathcal{D}$ using a reasonable number of samples?

The focus of this paper is the general case in which $\mathcal{D}$ is not necessarily a product distribution. Revenue maximization for product distributions was previously studied by [9, 11] who together achieve a polynomial-time constant-factor approximation, and by [5] who design a quasi-polynomial-time $(1 + \epsilon)$ approximation to the revenue of *deterministic auctions*. It is known, however, that the general case of correlated distributions is harder, e.g. deterministic prices can not provide a constant approximation [4].

The most natural approach for maximizing revenue for a distribution $\mathcal{D}$ given as a black box would be to sample some polynomial number of valuations from it, construct a revenue maximizing auction for this sample, and hope that the constructed auction also has good revenue on the original distribution $\mathcal{D}$. This, however, may fail terribly even for symmetric product distributions as shown in Proposition 3.

The astute reader will recognize this failure as a classic case of over-fitting: the optimal mechanism for the sample is so specifically targeted to the sample that it loses any optimality for the real distribution $\mathcal{D}$. The remedy for such over-fitting is well known: we need to "discourage" such tailoring and encourage "simple" auctions. At this point we need to specify what "simple" auctions are, a question closely related to how auctions are *represented*. The simplest answer is the "menu-complexity" suggested in [14]: we measure the complexity as the number of possible allocations of the auction, i.e. as the number of entries in the menu specifying the auction. More complex representations that may be more succinct for some auctions are also possible. Our lower bounds will apply to any *auction representation language*. Formally, an auction representation language is an arbitrary function that maps binary strings to auctions. The *complexity of an auction* in a given language is just the length of the smallest binary string that is mapped to this auction. Thus for example the menu-size of an auction corresponds to complexity in the representation language where the menu-entries are explicitly represented by listing the probabilities and price for each entry separately.[4]

Our Results

Fixing an auction representation language, we consider the following sample-and-optimize template for revenue maximization that takes into account the complexity of the output auction (in said representation language). The same idea is used in the context of prior-free mechanism design by [3]. In our setting, beyond the dependence on complexity of the auction, the number of samples needs to depend (polynomially) on three other parameters: the number of items m, the required precision ϵ, and the range of the values. Specifically, assume that all valuations in the support of $\mathcal{D}$ lie in the bounded range $1 \leq v(j) \leq H$ for all j. Since most of the expected value of a valuation

[4] Counting bits, we are off from just counting the number of menu entries by a factor of $O(m \log \epsilon)$ where ϵ is the precision in which real numbers are represented. We will focus on exponential versus polynomial complexities, so do not assign much importance to this gap.

may come from events whose probability is $O(1/H)$, it is clear that we will need $\Omega(H)$ samples to even notice these events.[5]

Sample-and-Optimize Algorithm Template

1. Sample $t = poly(C, m, \epsilon^{-1}, H)$ samples from $\mathcal{D}$.
2. Find an auction of complexity at most C that maximizes (as much as possible) the revenue for the uniform distribution over the sample and output it.

To convert this template to an algorithm, one must specify C and show how to compute an auction of complexity C which achieves high revenue on the sample. (Note that without the complexity bound, even fully maximizing revenue is efficiently done using the basic linear program, but this does not carry-over to complexity-bounded maximization.) Once the complexity is bounded, at least information-theoretically, the "usual" learning-like uniform convergence bounds indeed apply and we have:

Proposition 1. *Fix any auction representation language, and take an algorithm that follows this template and always produces an auction that approximates to within an α factor the optimal revenue from the* sample *over all auctions of complexity C. Then the produced auction also approximates to within a factor of $(1-\epsilon)\alpha$ the revenue from the real prior distribution $\mathcal{D}$ over all auctions of complexity C. (see full version)*

Thus we get approximate revenue maximization over all complexity-C auctions. However, if this limited class is inferior to general auctions, this does not yield approximate revenue maximization over all auctions, which is our goal. The following questions thus remain:

1. What is the complexity C required of an auction in order to obtain good revenue? What approximation can we get when we require C to be polynomial in H and m?
2. What is the computational complexity of step 2 of the algorithm template, i.e. of constructively finding a mechanism that maximizes revenue over mechanisms of bounded complexity C?

We provide definitive answers to the first question and preliminary answers to the second. Apriori, it is not even clear that any finite complexity C suffices for getting good revenue (for fixed H and m). Previous work ([12]) implies that arbitrarily good approximations are possible using menu-size complexity that is polynomial in H and exponential in m and for the special case $m = 2$ even poly-logarithmic size in H, [14]. This is done by taking the optimal auction and "rounding" its entries. This is trickier than it may seem since a slight change of probabilities may cause a great change in revenue, so such proofs need to carefully adjust the rounding of probabilities and prices making sure that significant revenue is never lost.[6] We describe a general way to perform this adjustment, allowing us to tighten these results: we get poly-logarithmic

[5] This also explains why bounding the range of valuations, as we do, is required for the sampling question to make any sense. Equivalently, we could have instead bounded the variance of $\mathcal{D}$ without significantly changing any of our results.

[6] Technically, there is no countable ϵ-net of auctions in the sense of approximating the revenue for every distribution. Instead, one needs to construct a "one-sided" net.

dependence in H for general m, stronger bounds for "monotone" valuations, and do it all effectively in a computational sense. Monotone valuations are restricted to have $v(1) \leq v(2) \leq ... \leq v(m)$, and naturally model cases such as values for a sequence of ad-slots or for increasing numbers of items in a multi-unit auction.

Theorem 1. *For every distribution $\mathcal{D}$ and every $\epsilon > 0$ there exists an auction with menu-size complexity at most $C = \left(\frac{\log H + \log m + \log \epsilon^{-1}}{\epsilon}\right)^{O(m)}$ whose revenue is at least $(1-\epsilon)$ fraction of the optimal revenue for $\mathcal{D}$. For the special case of distributions over "monotone" valuations, menu-size complexity of at most $C = m^{O((\log^3 H + \log^2 \epsilon)/\epsilon^2)}$ suffices. Furthermore, in both cases these auctions can be computed in $poly(C, H, \epsilon^{-1}, m)$ time by sampling $poly(C, H, \epsilon^{-1}, m)$ valuations from $\mathcal{D}$.*

So in general a menu of complexity exponential in m suffices. A basic question is whether complexity polynomial in m suffices (in terms of menu-size or perhaps other stronger representation languages). The "usual tricks" suffice to show that an $O(\log H)$-approximation of the revenue is possible with small menus.

Proposition 2. *For every distribution $\mathcal{D}$ there exists an auction with m menu-entries (and with all numbers represented in $O(\log H)$ bits) that extracts $\Omega(1/\log H)$ fraction of the optimal revenue from $\mathcal{D}$. Furthermore, this auction can be computed in polynomial time from $\mathrm{poly}(H)$ samples. (see full version)*

Can this be improved? Can we get a constant factor approximation with polynomial-size complexity? Our main result is negative. Previous techniques that separate the revenue of simple auctions from that of general auctions do not suffice for proving an impossibility here for two reasons. First, these bounds only apply to menu-size complexity and not to general auction representations; this is explicit in [14] and implicit in [4].[7] Second, these bounds proceed by giving an upper bound to the revenue that a single menu-entry can extract. Since small menus can extract an $O(1/\log H)$ fraction of the optimum revenue, such techniques can have no implications for sampling complexity since, as mentioned above, $\Omega(H)$ is a trivial lower bound on the sampling complexity. Our main result shows that even for a small range of values H, auctions may need to be exponentially complex in m in order to break the $O\left(\frac{1}{\log H}\right)$ barrier.

Theorem 2. *For every auction representation language and every $1 < H < 2^{m/400}$ there exists a distribution $\mathcal{D}$ on $[1..H]^m$ such that every auction with complexity at most $2^{m/400}$ has revenue that is at most an $O\left(\frac{1}{\log H}\right)$ fraction of the optimal revenue for $\mathcal{D}$.*

Notice that this immediately implies a similar exponential lower bound on the number of samples needed: since we allow any auction description language, simply listing the sample is one such language for which the lower bound holds.

Next, we examine our second question, regarding the computational complexity of "fitting" an auction of low complexity to sampled data. We show that it is NP-hard to compute an approximately optimal auction of a specified menu size.[8]

[7] Since these papers exhibit an explicit distribution providing the separation, the optimal auction can always be specified in some language by just listing the few parameters of said distribution.

[8] Here, we expect stronger auction representation languages to only be harder to deal with.

Theorem 3. *Given as input a sample of valuations and a menu-size C, it is NP-hard to approximate the optimal menu with size C to within any factor better than $1 - \frac{1}{e}\frac{H-1}{H}$.*

This hardness result does not preclude a satisfactory answer to our original goal of effectively finding an auction that approximates the revenue also on the original distribution $\mathcal{D}$ since for that it suffices to find a "small" auction with good revenue on the sample, rather than the "smallest" one. Thus a bi-criteria approximation to step 2 suffices, and may be algorithmically easier: find a menu of size $\mathrm{poly}(C, m, H)$ which approximates the revenue of the best menu of size C over a given sample. Whether this bi-criteria problem can be solved in polynomial time is left as our first open problem.

Our second open problem concerns the question of structured distributions, specifically product distributions over item values studied in [5, 9, 11]. Proposition 1 implies that, as these distributions can be succinctly represented, polynomially many samples suffice for finding a nearly optimal auction for product distributions.[9] It is not clear, however, how this can be done algorithmically and whether the simple menu-size auction description language suffices for succinctly representing the (approximately) optimal auction. Constant factor approximation with small menu-size (even deterministic menus) follow from [9, 11], but a $(1 + \epsilon)$-approximation is still open.

2 Preliminaries

2.1 The Model

In the *single-buyer unit-demand mechanism design problem*, or the *pricing problem* for short, we assume that there are m "items" or "outcomes" $[m] = \{1, \ldots, m\}$, and a single risk-neutral buyer equipped with a valuation $v \in \mathbb{R}_+^m$. Additionally, we assume the existence of an additional outcome $*$ for which a buyer has value 0 – e.g. the outcome in which the player receives no item. We assume that v is drawn from a distribution $\mathcal{D}$ supported on some family of valuations $\mathcal{V} \subseteq \mathbb{R}_+^m$.

We adopt the perspective of an auctioneer looking to sell the items in order to maximize his revenue. After soliciting a bid $b \in \mathcal{V}$, the auctioneer chooses an *allocation*, namely a (partial) lottery $x \in \Delta_m = \left\{x \in \mathbb{R}_+^m : \sum_i x_i \leq 1\right\}$ over the items, and a *payment* $p \in \mathbb{R}_+$. Formally, the auctioneer's task is to design a *mechanism*, equivalently, an auction, (x, p), where $x : \mathcal{V} \to \Delta_m$ maps a player's reported valuation to a lottery on the m items, and $p : \mathcal{V} \to \mathbb{R}_+$ maps the same report to a payment. When each allocation in the range of x is a deterministic choice of an item, we say the mechanism (or auction) is *deterministic*, otherwise it is *randomized*.

To simplify our results, we usually assume that players' valuations lie in a bounded range. Specifically, we require that the support $\mathcal{V}$ of our distribution is contained in $[1, H]^m$, for some finite upper-bound H which may depend on the number of items being sold. Given this assumption, we restrict our attention without loss of generality to mechanisms with payment rules constrained to prices in $[1, H] \cup \{0\}$. Moreover, we assume without loss of generality that $p(v) = 0$ only if $x(p) = \mathbf{0}$.[10]

[9] This is directly implied when the item values have finite (polynomial) support; the techniques used in section 4.1 suffice for showing it in general.

[10] An optimal mechanism satisfying these two properties always exists for all the problems we consider.

Whereas our complexity results are independent of the representation of $\mathcal{D}$, our algorithmic results hold in the *black-box model*, in which the auctioneer is given sample access to $\mathcal{D}$, and otherwise knows nothing about $\mathcal{D}$ besides its support $\mathcal{V}$.

2.2 Truthfulness and Menus

We constrain our mechanism (x, p) to be *truthful*: i.e. bidding $b = v$ maximizes the buyer's utility $v \cdot x(b) - p(b)$. The well known characterization below reduces the design of such a mechanism to the design of a *pricing menu*.

Fact 1. *A mechanism (x, p) is truthful if and only if there is a menu $M \subseteq \Delta_m \times \mathbb{R}$ of allocation/price pairs such that $(x(v), p(v)) \in \operatorname{argmax}_{(x,p)\in M}\{v \cdot x - p\}$.*

We adopt the menu perspective through much of this paper, interchangeably referring to a mechanism (aka auction) and its corresponding menu M. When interpreting a menu M as a mechanism, we break ties in $v \cdot x - p$ in favor higher prices. When every allocation in the menu is a deterministic choice of an item, we call M an *item-pricing menu*, otherwise we call it a *lottery-pricing menu*.

We also require our mechanisms to be *individually rational*. To enforce this, we assume that $(\mathbf{0}, 0)$ is in every menu. As described in Section 2.1, we usually restrict valuations to $[1, H]^m$ and payments for non-zero lotteries to $[1, H]$. Therefore, we think of a menu as a subset of $\Delta_m \times [1, H]$, and include $(0, 0)$ implicitly.

2.3 Auction Complexity and Benchmarks

Given a mechanism M and valuation v, we use $\operatorname{Rev}(M, v)$ to denote the payment of a buyer with valuation v when participating in the mechanism. Given a distribution $\mathcal{D}$ over valuations, we use $\operatorname{Rev}(M, \mathcal{D}) = \mathbf{E}_{v\sim\mathcal{D}} \operatorname{Rev}(M, v)$ to denote the expected revenue generated by the mechanism when a player is drawn from distribution $\mathcal{D}$. We use $\operatorname{Rev}(\mathcal{D})$ to denote the supremum, over all mechanisms M, of $\operatorname{Rev}(M, D)$. When $\mathcal{M}$ is a family of mechanisms, we use $\operatorname{Rev}(\mathcal{M}, \mathcal{D})$ to denote $\sup_{M\in\mathcal{M}} \operatorname{Rev}(M, \mathcal{D})$.

Recall that a auction description language is just a mapping from binary strings to mechanisms. I.e. it is simply a way of encoding menus in binary strings. The representation complexity of an auction in such a language is simply the length of the shortest string that is mapped to it. For this paper, the only important property of auction description languages is that there are at most 2^C auctions of complexity C. (Of course, in applications we will also worry about its expressive power, its computational difficulty, etc.) The simplest auction description language allows describing an auction by directly listing its menu entries one by one. Each menu entry is composed of $m + 1$ numbers, and if all the numbers can be presented using $O(r)$ bits of precision, then the total complexity of a k-entry auction in this format is $O(kmr)$. In this paper we never need more than $r = O(\log m + \log H + \log \epsilon^{-1})$ bits of precision, so the gap between menu-size (the number of menu entries) and complexity using this language (the total number of bits used in such a description) is not significant.

We use $\operatorname{Rev}_k(\mathcal{D})$ to denote the maximum revenue of a mechanism with complexity at most k. When using menu-size complexity, we say M is a k-menu if $|M| \leq k$, and use $\mathcal{M}_k$ to denote the set of all k-menus, and $\mathcal{M}_\infty$ to denote the set of all menus.

3 Sampling vs. Auction Complexity

3.1 Over Fitting with Complex Auctions

In this subsection we will consider the basic sampling algorithm that makes a small number of samples and optimizes the auction for this sample.

Naive Sample-and-Optimize Algorithm

1. Sample $t = poly(m)$ samples from $\mathcal{D}$.
2. Find an auction that maximizes revenue for the uniform distribution over samples.

We will show that this does not work even for symmetric product distributions. Let $\delta > 0$ be some small constant and let $\mathcal{D}$ be the distribution on valuations where item values are chosen identically and independently at random as follows: with probability δ: $v(j) = 1$; with probability δ/m: $v(j) = 2$; and otherwise: $v(j) = 0$. We will show that optimizing for a sample may give very low revenue on $\mathcal{D}$ itself:

Proposition 3. *Take a polynomial-size sample, with high probability, there is an auction that is optimal for the sample and yet its revenue from $\mathcal{D}$ is $O(\delta)$. (see full version)*

3.2 Uniform Convergence over Simple Auctions

When we limit the "complexity" of the auction that our algorithm is allowed to produce to be significantly smaller than the sample size, we can guarantee that the produced auction approximately maximizes revenue for the original distribution $\mathcal{D}$. Since there can not be too many auctions of low complexity, this follows by a standard application of tail bounds and the union bound as in [3].

Sample-and-Optimize Algorithm Template

1. Sample $t = poly(C, m, \epsilon^{-1}, H)$ samples from $\mathcal{D}$.
2. Output an auction of complexity at most C that approximately maximizes the revenue for the uniform distribution over the sample.

Proposition 1. *Fix any auction representation language, and take an algorithm that follows this template and always produces an auction that approximates to within an α factor the optimal revenue from the* sample *over all auctions of complexity C. Then the produced auction also approximates to within a factor of $(1 - \epsilon)\alpha$ the revenue from the real prior distribution $\mathcal{D}$ over all auctions of complexity C. (see full version)*

4 Constructions of Simple Approximating Auctions

4.1 From Rounding Lotteries to "Rounding" Auctions

When trying to approximate a given auction, it is natural to simply round all entries in the menu and hope that this does not hurt the revenue significantly. As mentioned in the introduction, this is not trivial since tiny decreases in probabilities or tiny increases in price may be the "last straw" chasing away bidders that made knife's-edge choice of the entry. This subsection shows that, never the less, this may be done with a little further tweaking: once we have a good way to round lotteries we can "round" entire menus, losing an approximation factor that is polynomially related to the rounding error.

Several previous works (e.g. [3,15]) have considered discretization of pricing mechanisms, most of which operate on item prices and therefore admit simpler "covers" of the space of mecanisms. Recently, [8] applied some of these ideas to lottery pricing, though their lemma is not at the level of generality required for our purposes. In particular, if we use their lemma directly, the additive loss in revenue due to discretization could be ϵH, which cannot guarantee any multiplicative approximation ratio.

Definition 1. *Let V be a set of valuations, and L be a set of lotteries. We say that L ϵ-covers V if for every lottery $x \in \Delta_m$ there exists a lottery $\widetilde{x} \in L$ such that for every $v \in V$ we have that $x \cdot v \geq \widetilde{x} \cdot v \geq (1-\epsilon)x \cdot v - \epsilon$.*

Lemma 1. *Let L be a set of lotteries that ϵ-covers a set of valuations V, for every menu M there exists a menu $\widetilde{M}$ all of whose entries have lotteries in L and have prices represented in $O(\log \epsilon^{-1} + \log H + \log m)$ bits such that for every $v \in V$ $Rev(\widetilde{M}, v) \geq (1-\epsilon')Rev(M, v) - \epsilon'$, where $\epsilon' = O(\log H \sqrt{\epsilon})$. Moreover, if the calculation of $\widetilde{x}$ from x is efficient then so is the calculation of $\widetilde{M}$ from M. (see full version)*

4.2 Approximations for General Valuations

So at this point we know that we just need to worry about rounding lotteries. Once we round all values to a small number of discrete values, we will get a small number of lotteries. Unfortunately, we need to use both an additive and multiplicative approximation error: the multiplicative approximation error allows a large additive error when values are close to H; and the additive error saves us from having to approximate multiplicatively very small probabilities. Combining these two notions of error allows us to make do with $O(\log H/\epsilon)$ discrete levels of approximation.

Proposition 4. *Let R_ϵ be the set of real numbers containing zero and all integer powers of $(1-\epsilon)$ in the range $[\epsilon/(Hm), 1]$, and let L_ϵ be the set of lotteries all of whose entries are in R_ϵ. Then L_ϵ ϵ-covers the set of all valuations $v \in [1, H]^m$. Moreover, calculating $\widetilde{x}$ from x can be done efficiently. (see full version)*

Corollary 1. *There exists an ϵ-cover of the set of all valuations $v : \{1...m\} \to [1, H]$ whose size is $((\log m + \log H + \log \epsilon^{-1})/\epsilon)^m$.*

Corollary 2. *(Part I of theorem 1 from the introduction) For every distribution $\mathcal{D}$ on $[1, H]^m$ and every $\epsilon > 0$ there exists an auction with menu-size complexity at most $C = \left(\frac{\log H + \log m + \log \epsilon^{-1}}{\epsilon}\right)^{O(m)}$ whose revenue is at least $(1-\epsilon)$ fraction of the optimal revenue for $\mathcal{D}$. Furthermore, this auction can be computed effectively (in polynomial time in its size) from a sample of size $\text{poly}(C, m, H, \epsilon^{-1})$.*

Proof. Combining corollary 1 using $O((\epsilon/\log H)^2)$ in place of ϵ with lemma 1 we get the existence of mechanism with menu-size complexity of $((\log m + \log H)/\epsilon)^{O(m)}$ whose approximation error (both additive and multiplicative) is ϵ. The additive approximation error of the whole mechanism is subsumed by the multiplicative one since optimal revenue is at least 1. We now plug this family of low complexity mechanisms into proposition 1, and obtain the required result, in the information-theoretic sense.

To compute the actual menu, we solve the linear program on the sample (that is of size polynomial in C, thus exponential in m), obtain the optimal mechanism M for the sample, and then round it to a mechanism $\widetilde{M}$ that provides the required approximation for the sample and – since it is of the right complexity – also for the distribution.

4.3 Approximations for Monotone Valuations

In this section, we consider a limited class of valuations and show that for distributions over this class a much smaller complexity is needed. The class we consider fixes an order on items, without loss of generality, the order $1, \ldots, m$. A valuation $v \in [1, H]^m$ is *monotone* if $v_i \leq v_{i+1}$ for $i \in \{1, \ldots, m-1\}$. Monotone valuations are natural in contexts such as *multi-unit auctions*, where m identical goods are being sold, and an outcome corresponds to the number of goods allocated to the buyer. In this setting, v_i is the player's value for i goods. Monotone valuations then correspond to a free disposal assumption in multi-unit auctions.

Next we show that for monotone valuations, we can find a small ϵ-cover of all lotteries, which implies, using our "Rounding Lotteries to Rounding Auctions" paradigm, small complexity auctions that approximate revenue well for all monotone valuations. This family of auctions has menu-size complexity polynomial in m when ϵ and H are constant, and quasi-polynomial when H is polynomial in m (and ϵ poly-logarithmic).

Theorem 4. *(Part II of theorem 1 from the introduction) If $\mathcal{D}$ is supported on monotone valuations then for every $\epsilon > 0$ there exists a menu M with $C = m^{O(\frac{\log^3 H + \log^2 \epsilon^{-1}}{\epsilon^2})}$ entries (and with all numbers with $O(\log m + \log H + \log \epsilon^{-1})$ bits of precision) such that $Rev(M, \mathcal{D}) \geq (1 - \epsilon) Rev(\mathcal{D})$. Furthermore, this auction can be computed in $poly(C, H, \epsilon^{-1}, m)$ time by sampling $poly(C, H, \epsilon^{-1}, m)$ valuations from $\mathcal{D}$.*

As in the proof of corollary 2, using lemma 1 and proposition 1, this theorem follows from the following lemma (proof in full version):

Lemma 2. *For every ϵ, there is a set of lotteries L whose size is $m^{O((\log H + \log \epsilon^{-1})/\epsilon)}$, that ϵ-covers the set of all monotone valuations with $v : \{1...m\} \to [1, H]$. Moreover, calculating $\widetilde{x}$ from x can be done efficiently.*

5 Lower Bound for Auction Complexity

Before we embark on our lower bound, we note the matching upper bound.

Proposition 2. *For every distribution $\mathcal{D}$ there exists an auction with m menu-entries (and with all numbers represented in $O(\log H)$ bits) that extracts $\Omega(1/\log H)$ fraction of the optimal revenue from $\mathcal{D}$. Furthermore, this auction can be computed in polynomial time from $\mathrm{poly}(H)$ samples. (see full version)*

Theorem 2. *For every auction representation language and every $1 < H < 2^{m/400}$ there exists a distribution $\mathcal{D}$ on $[1..H]^m$ such that every auction with complexity at most $2^{m/400}$ has revenue that is at most an $O\left(\frac{1}{\log H}\right)$ fraction of the optimal revenue for $\mathcal{D}$.*

Proof. We will construct the distribution $\mathcal{D}$ probabilistically. Our starting point will be a fixed baseline distribution $\mathcal{B}$ that takes an "equal revenue" one-dimensional distribution and spreads it symmetrically over a random subset of the items. Each valuation in the support of $\mathcal{B}$ is specified by a set $S \subset \{1..m\}$ of size exactly $k = m/3$ and an integer scale value $1 \leq z \leq \log H$. The valuation will give value 2^z for every item in S and value 1 for every other item. The probability distribution over these is induced by choosing S uniformly at random among sets of size k and choosing z as to obtain an "equal revenue distribution" $Pr[z = x] = 2^{-x}$. We can view this distribution over the v's as choosing uniformly at random from a multi-set V of exactly $\binom{m}{k}(H-1)$ valuations (for every set S of size k we have $H/2$ copies of a valuation with value 2, $H/4$ copies of a valuation with value 4 ... and a single copy of a valuation with value H). The point is that due to symmetry, it can be shown that, just like in the corresponding single dimensional case, $\text{Rev}(\mathcal{B})$ is constant (despite the expected value being $O(\log H)$). This is proven formally in lemma 3 below.

Taking the point of view of $\mathcal{B}$ being a random choice of a valuation from the multi-set V, we will now construct our distribution $\mathcal{D}$ as being a uniform choice over a random subset V' of V, where V' is of size $|V'| = K = 2^{m/100}$. We will now be able to provide two estimates. On one hand, since V' is a random sample from V, we expect that every fixed mechanism will extract approximately the same revenue from $\mathcal{D}$ as from $\mathcal{B}$. This can be shown to hold, w.h.p., simultaneously for *all* mechanisms in a small enough family and thus all mechanisms with sub-exponential complexity can only extract constant revenue. On the other hand, as V' is sparse, w.h.p. it does not contain two valuations whose subsets have a large intersection. This will suffice for extracting at east half of the expected value as revenue, an expected value that is $O(\log H)$. Lemma 4 below proves the former fact and lemma 6 below proves the latter.

Lemma 3. *Let $\mathcal{B}$ denote the distribution above then* $\text{Rev}(\mathcal{B}) \leq 2$.

Proof. We will prove this by reduction the the single dimensional case where Myerson's theorem can be used. Let us define the single dimensional distribution $\mathcal{G}$ that gives value 2^x with probability 2^{-x} for $x \in \{1... \log H\}$. The claim is that $Rev(\mathcal{B}) \leq \text{Rev}(\mathcal{G}) \leq 2$. Since $\mathcal{G}$ is single dimensional, the second inequality follows from Myerson's result stating that a single price mechanism maximizes revenue, as it is easy to verify that every possible single price gives revenue of at most 2.

To prove the first claim we build a single-parameter mechanism for $\mathcal{G}$ with the same revenue as a given multidimensional one for $\mathcal{B}$. Given a value 2^x distributed as in $\mathcal{G}$, our mechanism chooses a random subset S of size k and constructs a valuation v by combining the given single-parameter value with this set, so now v is distributed according to $\mathcal{B}$. We run the mechanism that was given to us for $\mathcal{B}$ and when it returns an lottery $a = a(v)$, we sell the item in the single parameter auction with probability $\sum_{j \in S} a_j$, asking for the same payment as asked for the lottery a in the $\mathcal{B}$-auction. Now notice that the same outcome that maximizes utility for v also maximizes utility for the single parameter buyer.

Lemma 4. *Let $\mathcal{M}$ be the set of mechanisms with complexity at most $2^{m/400}$ and choose the distribution $\mathcal{D}$ as described above then, w.h.p.,* $\text{Rev}(\mathcal{M}, \mathcal{D}) \leq 3$.

Proof. As $|M| \leq 2^{2^{m/400}}$, it follows from the following lemma and union bound.

Lemma 5. *Fix some mechanism M and choose the distribution $\mathcal{D}$ as described above then $Pr[\text{Rev}(M, \mathcal{D}) > 3] \leq exp(-K/H^2) \leq 2^{-2^{m/300}}$.*

Proof. Let $r_M(v)$ be the revenue that M extracts on valuation v. By definition $Rev(M, \mathcal{B}) = E_{v \in V} r_M(v)$ while $Rev(M, \mathcal{D}) = E_{v \in V'} r_M(v)$ (where the distribution is uniform over the multi-sets V and V' respectively). Lemma 3 bounded the former: $Rev(M, \mathcal{B}) \leq 2$. Since we are choosing V' to be a random multi-set of size K and since for all v we have $0 \leq r_M(v) \leq H$ then we can use Chernoff bounds to bound the probability that the expectation of $r_M(v)$ over the sample V' is larger than its expectation over the population V to be $Pr[|E_{v \in V'} r_M(v) - E_{v \in V} r_M(v)| > 1] \leq exp(-K/H^2)$.

Lemma 6. *Choose the distribution $\mathcal{D}$ as described above then, w.h.p,* $\mathrm{Rev}(\mathcal{D}) \geq \log H/2$.

Proof. We will use the following property that holds, w.h.p., for V': for every two different valuations in V' the sets of items S and T associated with them satisfy $|S \cap T| < m/6$. The reason that this property holds is that for any fixed T, since S is a random set of size $m/3$ the probability that $|S \cap T|/|T| \geq 1/2$ is $exp(-|T|) \leq 2^{-m/40}$. Now we can take a union bound over all $K^2 = 2^{m/50}$ possible pairs of S and T.

Using this property of $\mathcal{D}$ here is a mechanism that extracts as revenue at least half of the expected value of v, i.e. at least $(\log H)/2$ revenue: we have a menu entry for each element $v \in V'$. For v that gives value 2^z to the set S, this entry will offer every item in S with probability $1/|S| = 3/m$, and will ask for payment of 2^{z-1} for this lottery. Clearly if v chooses this entry it gets net utility of exactly 2^{z-1}, we need to show that the net utility from any other menu entry is less than this. Observe that v's value from a menu entry that corresponds to a set T is exactly $2^z \cdot |S \cap T|/|T|$ which due to our property is bounded from above by 2^{z-1}.

6 Computational Complexity

In this section, we examine the computational complexity of the algorithmic task associated with Proposition 1 when valuations lie in $[1, H]^m$. We restrict our attention to the menu-complexity model. The computational bottleneck is Step 2 of the algorithm, which computes a C-menu maximizing revenue for the uniform distribution over samples. Specifically, it requires the solution of the following optimization problem $MAXREV$. An instance of $MAXREV$ is given by an integer C and a sample $X = \{v_1, \ldots, v_n\} \subseteq [1, H]^m$. Feasible solutions of $MAXREV$ are menus with at most C entries, and the objective is to maximize revenue for a buyer drawn uniformly from X. We leave essentially open the exact computational complexity of approximating $MAXREV$, yet make some progress by showing the problem APX-hard.

Theorem 3. *Given as input a sample of valuations and a menu-size C, it is NP-hard to approximate the optimal menu with size C to within any factor better than* $1 - \frac{1}{e}\frac{H-1}{H}$.

Proof. We reduce from a promise problem of the NP-hard optimization problem *max cover*. For convenience, we use the equivalent *hitting set* formulation of max cover. The input is a family $\mathcal{S} = \{S_1, \ldots, S_n\}$ of subsets of $[m]$, and an integer k, and the output is a "hitting set" $T \subseteq [m]$ of size at most k maximizing the number of sets $S \in \mathcal{S}$ with which T has a non-empty intersection — we say those sets S are "hit" by T. We use the fact that it is NP-hard to distinguish between instances of hitting set in which the optimal solution hits all sets in $\mathcal{S}$, and instances in which the optimal solution hits less than a $1 - \frac{1}{e} + \epsilon$ fraction of the sets in $\mathcal{S}$, for any constant $\epsilon > 0$ (see Feige [13]).

Given an instance $(\mathcal{S}, k)$ of hitting set, we produce an instance (X, C) of $MAXREV$ as follows. We let $C = k$, and for each $S_i \in \mathcal{S}$ we include a valuation $v_i \in X$

such that $v_i(j) = H$ for $j \in S_i$, and $v_i(j) = 1$ otherwise. If there is a hitting set T of size k which hits every $S_i \in \mathcal{S}$, then the item-pricing C-menu $\{(e_j, H) : j \in T\}$, which prices every item $j \in T$ at H, generates a revenue of H from every valuation $v_i \in X$. On the other hand, we show that if there is a C-menu with average revenue at least $R = H - (\frac{1}{e} - \epsilon)(H-1)$ over X, then there is a hitting set of size k hitting at least a $\frac{R-1}{H-1} = 1 - \frac{1}{e} + \epsilon$ fraction of the sets in $\mathcal{S}$. Consider such a C-menu $M = \{(x_1, p_1), \ldots, (x_C, p_C)\}$, and draw an item j_t from each lottery x_t in M.[11] Let $T = \{j_1, \ldots, j_C\}$ be the resulting random hitting set of size $\leq C = k$. It suffices to show that T hits at least an $\frac{R-1}{H-1}$ fraction of the sets in $\mathcal{S}$ in expectation.

$$
\begin{aligned}
R = \operatorname*{avg}_{i=1}^{n} \mathrm{Rev}(M, v_i) &\leq \operatorname*{avg}_{i=1}^{n} \max_{t=1}^{C} v_i \cdot x_t && (\text{ by individual rationality }) \\
&= \operatorname*{avg}_{i=1}^{n} \max_{t=1}^{C} [H \cdot x_t(S_i) + 1 \cdot x_t([m] \setminus S_i)] && (\ x_t(S) \text{ denotes } \textstyle\sum_{j \in S} x_t(j)\ ) \\
&\leq \operatorname*{avg}_{i=1}^{n} \max_{t=1}^{C} [H \cdot x_t(S_i) + 1 - x_t(S_i)] && (\text{ because } \textstyle\sum_j x_t(j) \leq 1\ ) \\
&= 1 + (H-1) \operatorname*{avg}_{i=1}^{n} \max_{t=1}^{C} x_t(S_i) \\
&= 1 + (H-1) \operatorname*{avg}_{i=1}^{n} \max_{t=1}^{C} \mathbf{Pr}[j_t \in S_i] && (\text{ because } j_t \sim x_t\ ) \\
&\leq 1 + (H-1) \operatorname*{avg}_{i=1}^{n} \mathbf{Pr}[T \cap S_i \neq \emptyset]
\end{aligned}
$$

The reader might have noticed that step 2 of the 'Sample-and-Optimize' algorithm template, requiring the solution of an instance of $MAXREV$, is too restrictive as stated. An auctioneer may constrain the complexity of his sought mechanism either because it is believed that such a mechanism is approximately optimal [12], or because computational and/or practical considerations limit the auctioneer to only "simple" mechanisms. In both cases, it is perhaps more natural to seek a *bicriteria guarantee*: a mechanism of complexity polynomial in C, m, and $\log H$ which nevertheless approximates the revenue of the best mechanism of complexity C.

To illustrate this idea, consider the following variant of the 'Sample-and-Optimize' algorithm with step 2 replaced by its *bicriteria* version:

1. Sample $t = poly_1(C, m, \epsilon^{-1}, H)$ samples from $\mathcal{D}$;
2. Find an auction of complexity at most $poly_2(C, m, \epsilon^{-1}, \log H)$ that approximates the optimal auction of complexity C on the t samples;

Using the same idea in the proof of Prop 1, we can see that, in order to avoid overfitting for auctions of complexity at most $poly_2(C, m, \epsilon^{-1}, \log H)$, a sample size of $poly_1(C, m, \epsilon^{-1}, H)$ suffices, as long as $poly_1$ is a larger polynomial than $poly_2$. An α-approximation for the bicriteria $MAXREV$ problem implies one in the *black-box* model. We leave the approximability of the bicriteria $MAXREV$ as an open question.

[11] Since our lotteries are partial (i.e. $\sum_j x_t(j) \leq 1$) some of the items j_t may be the "null" item.
[12] For product distributions, such belief can be formally proved.

References

1. Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 512–521, 2011.
2. Saeed Alaei, Hu Fu, Nima Haghpanah, Jason Hartline, and Azarakhsh Malekian. Bayesian optimal auctions via multi-to single-agent reduction. In *ACM Conference on Electronic Commerce*, page 17, 2012.
3. M-F Balcan, Avrim Blum, Jason D Hartline, and Yishay Mansour. Mechanism design via machine learning. In *Foundations of Computer Science*, pages 605–614, 2005.
4. Patrick Briest, Shuchi Chawla, Robert Kleinberg, and S. Matthew Weinberg. Pricing randomized allocations. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 585–597, 2010.
5. Yang Cai and Constantinos Daskalakis. Extreme-value theorems for optimal multidimensional pricing. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 522–531, 2011.
6. Yang Cai, Constantinos Daskalakis, and S Matthew Weinberg. An algorithmic characterization of multi-dimensional mechanisms. In *Proceedings of the ACM Symposium on Theory of Computing*, pages 459–478, 2012.
7. Yang Cai, Constantinos Daskalakis, and S Matthew Weinberg. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 130–139, 2012.
8. Yang Cai and Zhiyi Huang. Simple and nearly optimal multi-item auctions. In *SODA*, pages 564–577, 2013.
9. Shuchi Chawla, Jason D. Hartline, and Robert D. Kleinberg. Algorithmic pricing via virtual valuations. In *ACM Conference on Electronic Commerce*, pages 243–251, 2007.
10. Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the 42nd ACM Symposium on Theory of computing*, pages 311–320, 2010.
11. Shuchi Chawla, David L. Malec, and Balasubramanian Sivan. The power of randomness in bayesian optimal mechanism design. In *Proceedings of the 11th ACM conference on Electronic commerce*, 2010.
12. Constantinos Daskalakis and Seth Matthew Weinberg. Symmetries and optimal multi-dimensional mechanism design. In *ACM Conference on Electronic Commerce*, pages 370–387, 2012.
13. Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, July 1998.
14. Sergiu Hart and Noam Nisan. The menu-size complexity of auctions. In *ACM Conference on Electronic Commerce*, page 565, 2013.
15. Jason D. Hartline and Vladlen Koltun. Near-optimal pricing in near-linear time. In *Proceedings of the 9th International Conference on Algorithms and Data Structures*, WADS'05, pages 422–431, Berlin, Heidelberg, 2005. Springer-Verlag.
16. Alejandro M. Manelli and Daniel R. Vincent. Bundling as an optimal selling mechanism for a multiple-good monopolist. *Journal of Economic Theory*, 127(1):1 – 35, 2006.
17. R. Preston McAfee and John McMillan. Multidimensional incentive compatibility and mechanism design. *Journal of Economic Theory*, 46(2):335 – 354, 1988.
18. Roger Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.
19. John Thanassoulis. Haggling over substitutes. *Journal of Economic Theory*, 117(2):217 – 245, 2004.