Revenue maximization via machine learning with noisy data

Anonymous Author(s)
Affiliation
Address
email

Abstract

Increasingly, copious amounts of consumer data are used to learn high-revenue mechanisms via machine learning. Existing research on mechanism design via machine learning assumes that there is a distribution over the buyers' values for the items for sale and that the learning algorithm's input is a *training set* sampled from this distribution. This setup makes the strong assumption that no noise is introduced during data collection. In order to help place mechanism design via machine learning on firm foundations, we investigate the extent to which this learning process is robust to noise. Optimizing revenue using noisy data is challenging because revenue functions are extremely volatile: an infinitesimal change in the buyers' values can cause a steep drop in revenue. Nonetheless, we provide guarantees when arbitrarily correlated noise is added to the training set; we only require that the noise's magnitude is bounded. We conclude with an application of our guarantees to *multi-task* mechanism design, where there are multiple distributions over buyers' values and the goal is to learn a high-revenue mechanism per distribution. To our knowledge, we are the first to study mechanism design via machine learning with noisy data as well as multi-task mechanism design.

1 Introduction

Revenue maximization in multi-item settings is one of the most important, long-standing open problems in mechanism design. In Bayesian Mechanism Design, there is a set of items for sale and an underlying distribution defining a set of agents' values for the items. A mechanism determines which buyers receive which items and what they pay. For decades, research in economics has assumed that the mechanism designer must know the exact distribution over buyers' values. An explosion of research [e.g., 1, 4, 17, 18, 21, 22, 24, 26, 27, 29, 32, 33, 37–41, 44] has relaxed this strong assumption: instead, the distribution is unknown and the mechanism designer only has a training set of i.i.d. samples. Using the training set, the goal is to learn a mechanism with high expected revenue. *Learning-based* mechanism design is on the verge of taking over as the main tool for designing high-revenue mechanisms for selling items—a cornerstone of many modern enterprises.

Motivated by recent literature on the brittleness of deep learning models in the face of *imperceptible* noise [28, 34, 45], an important question is whether adversarial noise has the same effect in learning-based mechanism design. Learning methods deployed in real-world settings to design mechanisms must be robust to minute noise in the data. We provide guarantees for learning mechanisms with high expected revenue in the face of an adversary that can add arbitrarily correlated noise to the training set; we only require that the noise's magnitude be bounded. In contrast, classic learning-theoretic results typically rely on the stronger assumption that the training samples are independent.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

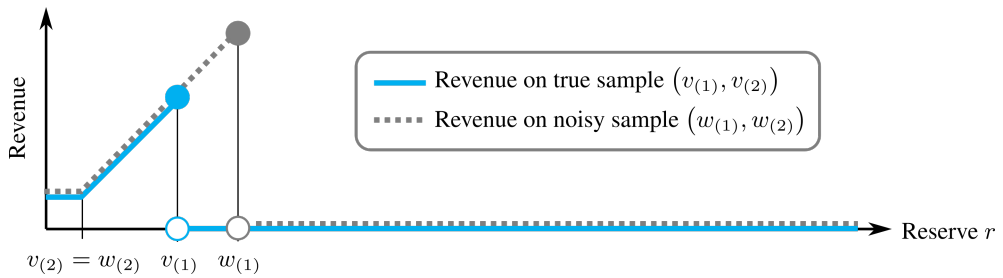

Figure 1: We illustrate revenue function volatility using the single-item second-price auction with a reserve r. Let $v_{(1)}$ be the highest bid and $v_{(2)}$ be the second-highest bid. Revenue as a function of the reserve r equals $v_{(2)}$ if $r \leq v_{(2)}$, r if $v_{(2)} < r \leq v_{(1)}$, and 0 if $r > v_{(1)}$, as illustrated by the blue solid line. Suppose we receive a noisy sample with highest bid $w_{(1)} > v_{(1)}$ and second-highest bid $w_{(2)} = v_{(2)}$. Revenue is illustrated by the dotted grey line. Setting the reserve equal to $w_{(1)}$ maximizes revenue on the noisy sample, but leads to zero revenue on the true values $(v_{(1)}, v_{(2)})$.

1.1 Summary of main contributions and overview of techniques

We set out to determine whether imperceptible adversarial noise in the training set can cause a catastrophic loss in the learned mechanism's expected revenue. Our main contribution is a *sensitivity analysis* of revenue with respect to the noise's magnitude, which answers this question in the negative. Our results apply to *empirical revenue maximization*, the canonical approach to learning-based mechanism design. To our knowledge, we are the first to study learning-based mechanism design under adversarial noise. We provide guarantees for optimizing revenue over several classic mechanism classes: second-price auctions with non-anonymous reserves under additive buyers, single-priced lottery mechanisms under unit-demand buyers, and item-pricing mechanisms under unit-demand buyers. These classes have been studied extensively [e.g., 6, 16, 20, 39, 44] and can be viewed as *hypothesis classes*, just as DNNs correspond to a particular (brittle) type of hypothesis class.

The key challenge we face is that revenue functions are extremely sensitive to small perturbations of the buyers' values. In a second-price auction, for example, slightly shifting the highest bid from below to above the reserve price can cause an arbitrarily large drop in revenue, as illustrated in Figure 1. The set of bidders whose bids are above their reserve prices can completely change when even an infinitesimal amount of noise is added, radically altering the analytical form of the revenue function. This is unlike most functions that we understand well from a learning-theoretic perspective, which generally are smooth, continuous, or—more broadly—exhibit a straightforward connection between parameters and output value. Despite the volatility of these revenue functions, we nonetheless are able to prove bounds on the revenue loss incurred by optimizing over a noisy training set.

Our second main contribution is an application of our guarantees to *multi-task mechanism design*. To our knowledge, we are the first to study this problem. The existing literature on mechanism design via machine learning assumes that there is a *single* distribution defining the buyers' values. Often, however, the mechanism designer may be interested in designing high-revenue mechanisms for multiple related distributions. Each distribution thus defines a distinct learning *task*. The goal is to leverage the similarity between the distributions to learn a high-revenue mechanism per distribution. Multi-task learning has proven useful in fields such as Computer Vision and Natural Language Processing [46], and we demonstrate its value in mechanism design as well. Our main technical insight is that we can transform training instances from one task into slightly noisy training instances for another task and make use of our near-optimality guarantees under noised samples.

1.2 Related research

A prominent line of research on machine learning in the presence of noise studies the case where some fraction of the training data may be noisy. Research in this vein includes settings where every instance can be corrupted with probability $\eta < 1$ [e.g., 3] and where an η-fraction of the training data can be corrupted [e.g., 14, 23]. Meanwhile, in our setting, every sample may be adversarially perturbed with probability 1, not just an η-fraction, though we require that the perturbation be bounded. Another line of research, especially prominent in Computer Vision, studies the case where the test data can

be adversarially perturbed within, for example, an ℓ_p-ball [28, 34, 45]. We, on the other hand, are concerned with noisy training data and face the unique challenges imposed by mechanism design.

To our knowledge, there is only one other paper with results that apply to mechanism design via machine learning when the samples are not necessarily drawn i.i.d. from the true distribution. Cai and Daskalakis [18] show that given access to a noisy distribution, it is possible to learn a mechanism with high expected revenue over the true distribution if the Kolmogorov distance between the noisy and true distributions are small. One might hope that the uniform distribution over the noisy training data could constitute the noisy distribution, but the Kolmogorov distance between this empirical distribution and the true distribution could be as large as 1, so these results do not apply to our setting.

In Section 3.4, our bounds improve based on how "nice" the distribution over buyers' values is, quantified by *dispersion* [5]. Dispersion has primarily been used to provide regret bounds [5, 8]. Balcan et al. [7] also use dispersion in mechanism design, though for a very different problem: estimating how much utility an agent can gain by misreporting their value in a manipulable mechanism.

The appendix summarizes additional related work on mechanism design with side information, multi-task learning, transfer learning, and approximate incentive compatibility.

2 Notation

There are n buyers and m items for sale. Each buyer $i \in [n]$ has a value for each item $j \in [m]$, denoted as $v_{ij} \geq 0$. We analyze unit-demand and additive buyers. If buyer i is unit-demand, he is only interested in obtaining one item, so his value for a bundle $b \subseteq [m]$ of goods is equal to the maximum value he has for any item in b, $\max_{j \in b} v_{ij}$. If buyer i is additive, his value for a bundle $b \subseteq [m]$ is $\sum_{j \in b} v_{ij}$. We use the notation $\boldsymbol{v}_i = (v_{i1}, \ldots, v_{im})$ to denote buyer i's values for all m items and $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) \in \mathbb{R}^{nm}$ to denote all n buyers' values. When there is only one item, we use the notation $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{R}^n$, where $v_i \in \mathbb{R}$ is buyer i's value for the item.

We study mechanism classes parameterized by vectors $\boldsymbol{r} \in \mathbb{R}^d$ for some d (for example, the class of second-price auctions parameterized by non-anonymous reserves $\boldsymbol{r}$). The mechanisms we analyze are dominant-strategy incentive compatible, so we assume the bids equal the buyers' true values. We denote the revenue of the mechanism defined by $\boldsymbol{r}$ when the buyers' values equal $\boldsymbol{v}$ by $\text{rev}_{\boldsymbol{r}}(\boldsymbol{v})$. For simplicity of notation, we assume $\text{rev}_{\boldsymbol{r}}(\boldsymbol{v})$ is in $[0, 1]$ though our results can be extended to the case where $\text{rev}_{\boldsymbol{r}}(\boldsymbol{v}) \in [0, H]$ for some H (all bounds must simply be multiplied by H).

3 Learning under adversarial noise

In this section, we study mechanism design via machine learning when the data is corrupted. Let $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\} \subset \mathbb{R}^{nm}_{\geq 0}$ be a set of valuation vectors drawn from an unknown distribution $\mathcal{D}$. Our learning algorithm receives a poisoned training set $\mathcal{S}' = \left\{\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}\right\} \subset \mathbb{R}^{nm}_{\geq 0}$. To model the *imperceptible* noise that corrupts the bids, we assume that the magnitude of the noise is bounded. We view this as a reasonable assumption as otherwise the noise in the bids would become discernible. In technical terms, for some $\epsilon > 0$, all samples $\ell \in [L]$, all bidders $i \in [n]$, and all items $j \in [m]$, $w^{(\ell)}_{ij} \in \left[v^{(\ell)}_{ij} - \epsilon, v^{(\ell)}_{ij}\right]$, or more succinctly, $\boldsymbol{v}^{(\ell)} - \boldsymbol{\epsilon} \leq \boldsymbol{w}^{(\ell)} \leq \boldsymbol{v}^{(\ell)}$. (If we only know that $\left\|\boldsymbol{v}^{(\ell)} - \boldsymbol{w}^{(\ell)}\right\|_\infty \leq \epsilon$, we can shift all bids in each vector $\boldsymbol{w}^{(\ell)}$ down by ϵ, in which case $\boldsymbol{v}^{(\ell)} - \boldsymbol{2\epsilon} \leq \boldsymbol{w}^{(\ell)} \leq \boldsymbol{v}^{(\ell)}$.) As is typical in the adversarial defense literature, we assume that ϵ is known. Our goal is to use the noisy set $\mathcal{S}'$ to learn a mechanism with high expected revenue over $\mathcal{D}$.

Our approach. We begin by fixing a mechanism class parameterized by vectors $\boldsymbol{r} \in \mathbb{R}^d$ for some d (for example, the class of second-price auctions parameterized by non-anonymous reserves $\boldsymbol{r}$). Given the training set $\mathcal{S}'$, the most widely used learning algorithm is *empirical revenue maximization (ERM)*, which returns the parameter setting $\hat{\boldsymbol{r}}'$ that maximizes average empirical revenue over $\mathcal{S}'$. This algorithm will be the subject of our study. Throughout this section, we analyze the following key question: what is the difference between the expected revenue of the mechanism defined by $\hat{\boldsymbol{r}}'$, $\mathbb{E}_{\boldsymbol{v} \sim \mathcal{D}}\left[\text{rev}_{\hat{\boldsymbol{r}}'}(\boldsymbol{v})\right]$, and that of the optimal mechanism in the class, $\max_{\boldsymbol{r} \in \mathbb{R}^d} \mathbb{E}\left[\text{rev}_{\boldsymbol{r}}(\boldsymbol{v})\right]$?

Another key aspect of learning besides optimality is sample complexity. In machine learning beyond the context of mechanism design, prior research [e.g., 42] has shown that a large increase in sample

complexity is needed to handle noise with bounded ℓ_∞-norm. By contrast, under the same noise assumption, we show that empirical revenue maximization can achieve near-optimal revenue *without significantly higher sample complexity*. This exposes a notable contrast between the two learning tasks. We provide lower bounds showing that ERM's loss has an optimal dependence on the noise ϵ.

3.1 Robustness

We begin by providing guarantees for any mechanism class that satisfies a notion of robustness, which helps us isolate exactly the properties we need to prove that robust revenue optimization is possible. A class is robust if it satisfies two properties. The first is a *stability* property. Let $\hat{\boldsymbol{r}}$ be the *empirically optimal* parameter setting over the set $\mathcal{S}$ of true samples, which means that of all $\boldsymbol{r} \in \mathbb{R}^d$, average revenue over $\mathcal{S}$ is maximized when $\boldsymbol{r} = \hat{\boldsymbol{r}}$. Let $\hat{\boldsymbol{r}}'$ be the empirically optimal parameter vector over the noisy set $\mathcal{S}'$. Stability is satisfied when the average revenues of $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{r}}'$ over $\mathcal{S}$ are close.

Our second property relies on a classic learning-theoretic notion of convergence. It is satisfied when, for every parameter vector $\boldsymbol{r}$, average revenue over the set $\mathcal{S}$ is close to expected revenue. Although convergence bounds have been derived in prior research for the mechanism classes we analyze [6, 39], a convergence bound alone *does not imply any guarantees* whatsoever for optimization with noisy data. This is why we introduce the separate notion of stability, which has not been previously studied in the automated mechanism design literature. The challenge then lies in proving that a variety of mechanism classes satisfy stability. To begin, we define the two properties formally as follows:

Definition 3.1. Given two functions $p : [0, 1] \times \mathbb{Z}^2 \to \mathbb{R}$ and $q : [0, 1] \times \mathbb{Z}^3 \to \mathbb{R}$, we say that a mechanism class is *p-stable and q-convergent* if the following conditions hold:

1. *p-stable.* For any $L \geq 1$, let $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\} \subset \mathbb{R}^{nm}_{\geq 0}$ and $\mathcal{S}' = \left\{\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}\right\} \subset \mathbb{R}^{nm}_{\geq 0}$ be two arbitrary sets of vectors such that $\boldsymbol{v}^{(\ell)} - \boldsymbol{\epsilon} \leq \boldsymbol{w}^{(\ell)} \leq \boldsymbol{v}^{(\ell)}$ for all $\ell \in [L]$. Let $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{r}}'$ be the empirically optimal parameter vectors over $\mathcal{S}$ and $\mathcal{S}'$: $\hat{\boldsymbol{r}} = \operatorname{argmax}_{\boldsymbol{r} \in \mathbb{R}^d} \sum_{\ell=1}^{L} \operatorname{rev}_{\boldsymbol{r}}\left(\boldsymbol{v}^{(\ell)}\right)$ and $\hat{\boldsymbol{r}}' = \operatorname{argmax}_{\boldsymbol{r} \in \mathbb{R}^d} \sum_{\ell=1}^{L} \operatorname{rev}_{\boldsymbol{r}}\left(\boldsymbol{w}^{(\ell)}\right)$. We require that on average over $\mathcal{S}$, the revenues of $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{r}}'$ are close: $\frac{1}{L}\sum_{\ell=1}^{L} \operatorname{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) - \operatorname{rev}_{\hat{\boldsymbol{r}}'}\left(\boldsymbol{v}^{(\ell)}\right) \leq p(\epsilon, n, m)$.

2. *q-convergent.* For any $L \geq 1$ and $\delta \in (0, 1)$, with probability $1 - \delta$ over the draw $\mathcal{S} \sim \mathcal{D}^L$, for every vector $\boldsymbol{r} \in \mathbb{R}^d$, the difference between the average revenue over $\mathcal{S}$ and the expected revenue is at most $q(\delta, L, n, m)$: $\left|\frac{1}{L}\sum_{\boldsymbol{v} \in \mathcal{S}} \operatorname{rev}_{\boldsymbol{r}}(\boldsymbol{v}) - \mathbb{E}\left[\operatorname{rev}_{\boldsymbol{r}}(\boldsymbol{v})\right]\right| \leq q(\delta, L, n, m)$.

If a mechanism class is p-stable and q-convergent, then the expected revenue of the empirically optimal mechanism over the noisy samples $\mathcal{S}'$ is close to the expected revenue of the optimal mechanism in the class. For completeness, the proof is in Appendix B.

Fact 3.2. *Fix a p-stable and q-convergent mechanism class. Let $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\}$ and $\mathcal{S}' = \left\{\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}\right\}$ be two sets of valuation vectors such that for all $\ell \in [L]$, $\boldsymbol{v}^{(\ell)} - \boldsymbol{\epsilon} \leq \boldsymbol{w}^{(\ell)} \leq \boldsymbol{v}^{(\ell)}$. Let $\hat{\boldsymbol{r}}'$ be empirically optimal over $\mathcal{S}'$: $\hat{\boldsymbol{r}}' = \operatorname{argmax}_{\boldsymbol{r} \in \mathbb{R}^d} \sum_{\ell=1}^{L} \operatorname{rev}_{\boldsymbol{r}}\left(\boldsymbol{w}^{(\ell)}\right)$. With probability $1 - \delta$ over $\mathcal{S} \sim \mathcal{D}^L$, $\max_{\boldsymbol{r} \in \mathbb{R}^d} \mathbb{E}_{\boldsymbol{v} \sim \mathcal{D}}\left[\operatorname{rev}_{\boldsymbol{r}}(\boldsymbol{v})\right] - \mathbb{E}_{\boldsymbol{v} \sim \mathcal{D}}\left[\operatorname{rev}_{\hat{\boldsymbol{r}}'}(\boldsymbol{v})\right] \leq p(\epsilon, n, m) + 2q(\delta, L, n, m)$.*

We now prove that several mechanism classes are stable: second-price auctions with non-anonymous reserves and lotteries. (For intuition, we also analyze the simpler class of second-price auctions with anonymous reserves in Appendix B.1.) Table 1 summarizes our results.

3.2 Non-anonymous second-price auctions

We prove that *second-price auctions with non-anonymous reserves* and additive bidders are robust. In the single-item setting, this auction is defined by a vector $\boldsymbol{r} \in \mathbb{R}^n$, where r_i is bidder i's *reserve price*. Each bidder submits a bid and the mechanism discards all bidders whose bids are smaller than their reserves. If there are bidders remaining, the highest bidder, say bidder i, wins and pays the maximum of the second-highest remaining bid and r_i (or r_i if there are no other remaining bids). In the multi-item case, there is only one copy of each item and there is a separate auction per item. The mechanism is defined by $\boldsymbol{r} = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_m) \in \mathbb{R}^{nm}$, where $\boldsymbol{r}_j \in \mathbb{R}^n$ is the reserve vector for item j.

We begin by analyzing single-item auctions, which then implies guarantees for multiple items. The key challenge is that the set of bidders whose bids are above their reserves can completely change when even infinitesimal noise is added, radically altering the analytical form of the revenue function.

Mechanism class	Buyers' values	p-stable	Error bound
Single-item anonymous second-price auction	Additive	$p(\epsilon, n, m) = 2\epsilon$ (Theorem B.2)	$\tilde{O}(\epsilon + \sqrt{1/L})$ (Corollary B.6)
Multi-item non-anonymous second-price auction	Additive	$p(\epsilon, n, m) = 2m\epsilon$ (Theorem 3.3)	$\tilde{O}(m\epsilon + \sqrt{nm/L})$ (Corollary 3.7)
Multi-item non-anonymous lotteries	Unit-demand	$p(\epsilon, n, m) = n\epsilon$ (Theorem 3.9)	$\tilde{O}(n\epsilon + \sqrt{nm/L})$ (Corollary 3.10)
Item-pricing mechanisms	Unit-demand	Does not satisfy (Lemma B.11)	$\tilde{O}(nm^2(\kappa\epsilon + \sqrt{1/L}))^*$ (Theorem 3.13, Lemma 3.15)

* The distribution over buyers' values is κ-bounded (Definition 3.14).

Table 1: Our p-stability guarantees together with the resulting error bounds. Item-pricing mechanisms do not satisfy p-stability, so we use alternative techniques to provide an error bound (Section 3.4).

Theorem 3.3. *Single-item non-anonymous second-price auctions are p-stable with $p(\epsilon, n, m) = 2\epsilon$.*

Proof. For any $L \geq 1$, let $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\} \subset \mathbb{R}^n_{\geq 0}$ and $\mathcal{S}' = \left\{\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}\right\} \subset \mathbb{R}^n_{\geq 0}$ be two sets of valuation vectors such that $\boldsymbol{v}^{(\ell)} - \boldsymbol{\epsilon} \leq \boldsymbol{w}^{(\ell)} \leq \boldsymbol{v}^{(\ell)}$ for all $\ell \in [L]$. Let $\hat{\boldsymbol{r}}'$ be the empirically optimal reserve vector over $\mathcal{S}'$ and let $\hat{\boldsymbol{r}}$ be empirically optimal over $\mathcal{S}$. This proof relies on two key lemmas. The first states that if we shift the reserve vector $\hat{\boldsymbol{r}}$—which is empirically optimal over $\mathcal{S}$—down by an additive factor of $\boldsymbol{\epsilon} = (\epsilon, \ldots, \epsilon) \in \mathbb{R}^n$ and apply it to the valuations in $\mathcal{S}'$, little revenue is lost. We use the standard notation $\langle x \rangle = \max\{x, 0\}$. The full proof is in Appendix B.2.

Lemma 3.4. *Let* $\boldsymbol{r}_\epsilon = \left(\langle \hat{r}_1 - \epsilon \rangle, \ldots, \langle \hat{r}_n - \epsilon \rangle\right)$. *If* $\text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) > 0$, *then* $\text{rev}_{\boldsymbol{r}_\epsilon}\left(\boldsymbol{w}^{(\ell)}\right) \geq \text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) - 2\epsilon$.

Proof sketch of Lemma 3.4. Suppose that $\text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) > 0$. Let i be the winner under the values $\boldsymbol{v}^{(\ell)}$ and reserve $\hat{\boldsymbol{r}}$. Since $v_i^{(\ell)} \geq \hat{r}_i$, we know that $w_i^{(\ell)} \geq \langle v_i^{(\ell)} - \epsilon \rangle \geq \langle \hat{r}_i - \epsilon \rangle$, so under the values $\boldsymbol{w}^{(\ell)}$ and reserve $\boldsymbol{r}_\epsilon$, there is at least one bidder whose bid is at least his reserve. Let i' be the winner under the valuation vector $\boldsymbol{w}^{(\ell)}$ and reserve $\boldsymbol{r}_\epsilon$. We split this proof into four cases that depend on:

1. Whether or not $i = i'$, and
2. Whether—under $\boldsymbol{v}^{(\ell)}$ and $\hat{\boldsymbol{r}}$—there is another bidder besides i whose bid is at least his reserve (in other words, whether or not $\{t : v_t^{(\ell)} \geq \hat{r}_t, t \neq i\} = \emptyset$).

In the first case, $i = i'$ and $\{t : v_t^{(\ell)} \geq \hat{r}_t, t \neq i\} \neq \emptyset$. Let k be the index of the second-highest bidder in $\boldsymbol{v}^{(\ell)}$ whose bid is above his reserve in $\hat{\boldsymbol{r}}$: $k = \text{argmax}_{t \neq i}\{v_t^{(\ell)} : v_t^{(\ell)} \geq \hat{r}_t\}$. This means that $\text{rev}_{\hat{\boldsymbol{r}}}(\boldsymbol{v}^{(\ell)}) = \max\{\hat{r}_i, v_k^{(\ell)}\}$. Since $v_k^{(\ell)} \geq \hat{r}_k$, we have that $w_k^{(\ell)} \geq \langle v_k^{(\ell)} - \epsilon \rangle \geq \langle \hat{r}_k - \epsilon \rangle$. Since $k \neq i$ and $i = i'$, it must be that $k \neq i'$, so $\{t : w_t^{(\ell)} \geq \langle \hat{r}_t - \epsilon \rangle, t \neq i'\} \neq \emptyset$ (in particular, the set contains k). Let k' be the index of the second-highest bidder in $\boldsymbol{w}^{(\ell)}$ whose bid is above his reserve in $\boldsymbol{r}_\epsilon$: $k' = \text{argmax}_{t \neq i'}\{w_t^{(\ell)} : w_t^{(\ell)} \geq \langle \hat{r}_t - \epsilon \rangle\}$, which means that $\text{rev}_{\boldsymbol{r}_\epsilon}(\boldsymbol{w}^{(\ell)}) = \max\{\hat{r}_{i'} - \epsilon, w_{k'}^{(\ell)}\}$. Since $k \in \{t : w_t^{(\ell)} \geq \langle \hat{r}_t - \epsilon \rangle, t \neq i'\}$, we know that $w_{k'}^{(\ell)} \geq w_k^{(\ell)} \geq v_k^{(\ell)} - \epsilon$. Putting this all together, we prove that $\text{rev}_{\boldsymbol{r}_\epsilon}(\boldsymbol{w}^{(\ell)}) \geq \max\{\hat{r}_i - \epsilon, w_{k'}^{(\ell)}\} \geq \max\{\hat{r}_i, v_k^{(\ell)}\} - 2\epsilon = \text{rev}_{\hat{\boldsymbol{r}}}(\boldsymbol{v}^{(\ell)}) - 2\epsilon$. We prove the other three cases in the appendix. □

We use Lemma 3.4 to continue Theorem 3.3's proof. Let I be the set of indices ℓ such that $\text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) > 0$. By Lemma 3.4, $\sum_{\ell=1}^{L} \text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) = \sum_{\ell \in I} \text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) \leq \sum_{\ell \in I} \text{rev}_{\boldsymbol{r}_\epsilon}\left(\boldsymbol{w}^{(\ell)}\right) + 2L\epsilon \leq \sum_{\ell=1}^{L} \text{rev}_{\boldsymbol{r}_\epsilon}\left(\boldsymbol{w}^{(\ell)}\right) + 2L\epsilon \leq \sum_{\ell=1}^{L} \text{rev}_{\hat{\boldsymbol{r}}'}\left(\boldsymbol{w}^{(\ell)}\right) + 2L\epsilon$.

Next, we prove that for any reserve vector $\boldsymbol{r}$, revenue under the samples $\boldsymbol{v}^{(\ell)}$ will only be higher than revenue under the samples $\boldsymbol{w}^{(\ell)}$, which intuitively makes sense since $\boldsymbol{w}^{(\ell)} \leq \boldsymbol{v}^{(\ell)}$. The proof, which has a similar structure as the proof of Lemma 3.4, is in Appendix B.2.

Lemma 3.5. *For all samples $\ell \in [L]$ and reserve vectors $\boldsymbol{r} \in \mathbb{R}^n$,* $\text{rev}_{\boldsymbol{r}}\left(\boldsymbol{w}^{(\ell)}\right) \leq \text{rev}_{\boldsymbol{r}}\left(\boldsymbol{v}^{(\ell)}\right)$.

Since $\sum_{\ell=1}^{L} \text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) \leq \sum \text{rev}_{\hat{\boldsymbol{r}}'}\left(\boldsymbol{w}^{(\ell)}\right) + 2L\epsilon$, Lemma 3.5 implies that the theorem holds. □

We next use Theorem 3.3 to prove stability guarantees in the case where there are multiple items and n additive buyers. It follows from the fact that the mechanism's revenue decomposes additively over the items. The proof is in Appendix B.2.

Theorem 3.6. *The set of multi-item non-anonymous second-price auctions under n additive buyers is p-stable with $p(\epsilon, n, m) = 2m\epsilon$.*

Theorem 3.6 and the fact that the class is q-convergent with $q(\delta, L, n, m) = O\left(\sqrt{\frac{1}{L}\left(nm\log(nm) + \log\frac{1}{\delta}\right)}\right)$ [39] implies that optimization under noise is possible:

Corollary 3.7. *Let $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\} \subset \mathbb{R}_{\geq 0}^{nm}$ and $\mathcal{S}' = \left\{\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}\right\} \subset \mathbb{R}_{\geq 0}^{nm}$ be two sets such that for all $\ell \in [L]$, $\boldsymbol{v}^{(\ell)} - \boldsymbol{\epsilon} \leq \boldsymbol{w}^{(\ell)} \leq \boldsymbol{v}^{(\ell)}$. Let $\hat{\boldsymbol{r}}' = \text{argmax}_{\boldsymbol{r} \in \mathbb{R}^{nm}} \sum_{\ell=1}^{L} \text{rev}_{\boldsymbol{r}}\left(\boldsymbol{w}^{(\ell)}\right)$. With high probability over $\mathcal{S} \sim \mathcal{D}^L$, $\max_{\boldsymbol{r} \in \mathbb{R}^{nm}} \mathbb{E}_{\boldsymbol{v} \sim \mathcal{D}}\left[\text{rev}_{\boldsymbol{r}}(\boldsymbol{v})\right] - \mathbb{E}_{\boldsymbol{v} \sim \mathcal{D}}\left[\text{rev}_{\hat{\boldsymbol{r}}'}(\boldsymbol{v})\right] = \tilde{O}\left(m\epsilon + \sqrt{\frac{nm}{L}}\right)$.*

This dependence on $m\epsilon$ is tight: no algorithm has better error than $m\epsilon$. Therefore, empirical revenue maximization provides an optimal dependence on the error term ϵ. The proof is in Appendix B.2.

Proposition 3.8. *Fix an arbitrary error term ϵ. For any deterministic algorithm $\mathcal{A}$ that takes as input a training set $\mathcal{S} \subseteq \mathbb{R}^{nm}$ and returns a vector of non-anonymous reserves $\mathcal{A}(\mathcal{S}) \in \mathbb{R}^{nm}$, there exists a distribution $\mathcal{D}$ such that with probability 1 over the draw $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\} \sim \mathcal{D}^L$, $\max_{\boldsymbol{r} \in \mathbb{R}^{nm}} \mathbb{E}\left[\text{rev}_{\boldsymbol{r}}(\boldsymbol{v})\right] - \mathbb{E}\left[\text{rev}_{\mathcal{A}(\mathcal{S}')}(\boldsymbol{v})\right] = \Omega(m\epsilon)$ for some noisy training set $\mathcal{S}' = \left\{\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}\right\}$ such that $\left\|\boldsymbol{v}^{(\ell)} - \boldsymbol{w}^{(\ell)}\right\|_\infty \leq \epsilon$ for all $\ell \in [L]$.*

3.3 Lotteries

We next prove that single-priced lottery mechanisms satisfy our stability and convergence conditions. We analyze a setting where there are n unit-demand buyers with values for m items. Unlike the previous section, we assume there are at least n units of each good available. A single-priced lottery is defined by $n(m+1)$ parameters: for each buyer i, there is a price $r_{i0} \in \mathbb{R}_{\geq 0}$ and a set of probabilities $r_{i1}, \ldots, r_{im} \in [0, 1]$ with $\sum_{j=1}^{m} r_{ij} = 1$. If the buyer chooses to pay r_{i0}, she will receive one item $J \in [m]$, and $\Pr[J = j] = r_{ij}$. Therefore, her expected utility is $\sum_{j=1}^{m} v_{ij} r_{ij} - r_{i0}$. She will choose to participate in the lottery so long as her expected utility is at least 0. We prove that this mechanism class satisfies our stability and convergence conditions. The full proof is in Appendix B.3.

Theorem 3.9. *The set of lotteries with n unit-demand buyers is p-stable with $p(\epsilon, n, m) = n\epsilon$.*

Proof sketch. We sketch the proof that $p(\epsilon, n, m) = \epsilon$ in the single-buyer setting ($n = 1$) and for the sake of generality, we prove the guarantee for n buyers in the appendix. For any $L \geq 1$, let $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\} \subset \mathbb{R}_{\geq 0}^{m}$ and $\mathcal{S}' = \left\{\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}\right\} \subset \mathbb{R}_{\geq 0}^{m}$ be two arbitrary sets of valuation vectors such that $\boldsymbol{v}^{(\ell)} - \boldsymbol{\epsilon} \leq \boldsymbol{w}^{(\ell)} \leq \boldsymbol{v}^{(\ell)}$ for all $\ell \in [L]$. Let $\hat{\boldsymbol{r}} = (\hat{r}_0, \hat{r}_1, \ldots, \hat{r}_m)$ be the empirically optimal parameter vector over the set $\mathcal{S}$ and let $\hat{\boldsymbol{r}}' = (\hat{r}'_0, \hat{r}'_1, \ldots, \hat{r}'_m)$ be the empirically optimal parameter vector over the set $\mathcal{S}'$. We prove that $\frac{1}{L}\sum_{\ell=1}^{L} \text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) - \text{rev}_{\hat{\boldsymbol{r}}'}\left(\boldsymbol{v}^{(\ell)}\right) \leq \epsilon$.

We first prove that if we shift the price $\hat{r}_0$ down by ϵ and evaluate the resulting lottery over $\mathcal{S}'$, little revenue is lost. Specifically, letting $\boldsymbol{r}_\epsilon = (\max\{\hat{r}_0 - \epsilon, 0\}, \hat{r}_1, \ldots, \hat{r}_n)$ we prove that if $\text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) > 0$, then $\text{rev}_{\boldsymbol{r}_\epsilon}\left(\boldsymbol{w}^{(\ell)}\right) \geq \text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) - \epsilon$. This implies that $\sum_{\ell=1}^{L} \text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) \leq \sum_{\ell=1}^{L} \text{rev}_{\boldsymbol{r}_\epsilon}\left(\boldsymbol{w}^{(\ell)}\right) + L\epsilon$. Since $\hat{\boldsymbol{r}}'$ is empirically optimal under $\mathcal{S}'$, $\sum_{\ell=1}^{L} \text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) \leq \sum_{\ell=1}^{L} \text{rev}_{\hat{\boldsymbol{r}}'}\left(\boldsymbol{w}^{(\ell)}\right) + L\epsilon$.

Next, we prove that for any parameter vector $\boldsymbol{r}$, revenue under the samples $\boldsymbol{v}^{(\ell)}$ will only be higher than revenue under the samples $\boldsymbol{w}^{(\ell)}$. Specifically, for every $\ell \in [L]$ and any parameter vector $\boldsymbol{r} \in \mathbb{R}^{m+1}$, $\text{rev}_{\boldsymbol{r}}\left(\boldsymbol{w}^{(\ell)}\right) \leq \text{rev}_{\boldsymbol{r}}\left(\boldsymbol{v}^{(\ell)}\right)$. This implies that $\sum_{\ell=1}^{L} \text{rev}_{\hat{\boldsymbol{r}}}\left(\boldsymbol{v}^{(\ell)}\right) \leq L\epsilon + \sum_{\ell=1}^{L} \text{rev}_{\hat{\boldsymbol{r}}'}\left(\boldsymbol{v}^{(\ell)}\right)$. □

By a natural generalization of prior research [6], $q(\delta, L, n, m) = O\left(\sqrt{\frac{1}{L}\left(nm\log(nm) + \log\frac{1}{\delta}\right)}\right)$. Fact 3.2 and Theorem 3.9 imply our main result for this section—that the class of lotteries is robust:

Corollary 3.10. *Let* $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\} \subset \mathbb{R}^{nm}_{\geq 0}$ *and* $\mathcal{S}' = \left\{\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}\right\} \subset \mathbb{R}^{nm}_{\geq 0}$ *be two sets such that for all* $\ell \in [L]$, $\boldsymbol{v}^{(\ell)} - \boldsymbol{\epsilon} \leq \boldsymbol{w}^{(\ell)} \leq \boldsymbol{v}^{(\ell)}$. *Let* $\hat{\boldsymbol{r}}' = \operatorname{argmax}_{\boldsymbol{r} \in \mathbb{R}^{n(m+1)}} \sum_{\ell=1}^{L} \operatorname{rev}_{\boldsymbol{r}}\left(\boldsymbol{w}^{(\ell)}\right)$. *With high probability over* $\mathcal{S} \sim \mathcal{D}^L$, $\max_{\boldsymbol{r} \in \mathbb{R}^{n(m+1)}} \mathbb{E}_{\boldsymbol{v} \sim \mathcal{D}}\left[\operatorname{rev}_{\boldsymbol{r}}(\boldsymbol{v})\right] - \mathbb{E}\left[\operatorname{rev}_{\hat{\boldsymbol{r}}'}(\boldsymbol{v})\right] = \tilde{O}\left(n\epsilon + \sqrt{\frac{nm}{L}}\right)$.

This dependence on $n\epsilon$ is tight: no algorithm has better error than $n\epsilon$. The proof is in Appendix B.3.

Proposition 3.11. *Fix an arbitrary error term* ϵ. *For any deterministic algorithm* $\mathcal{A}$ *that takes as input a training set* $\mathcal{S} \subseteq \mathbb{R}^{nm}$ *and returns a vector of lottery parameters* $\mathcal{A}(\mathcal{S}) \in \mathbb{R}^{n(m+1)}$, *there exists a distribution* $\mathcal{D}$ *such that with probability* 1 *over the draw* $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\} \sim \mathcal{D}^L$, $\max_{\boldsymbol{r} \in \mathbb{R}^{n(m+1)}} \mathbb{E}\left[\operatorname{rev}_{\boldsymbol{r}}(\boldsymbol{v})\right] - \mathbb{E}\left[\operatorname{rev}_{\mathcal{A}(\mathcal{S}')}(\boldsymbol{v})\right] = \Omega(n\epsilon)$ *for some noisy training set* $\mathcal{S}' = \left\{\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}\right\}$ *such that* $\left\|\boldsymbol{v}^{(\ell)} - \boldsymbol{w}^{(\ell)}\right\|_\infty \leq \epsilon$ *for all* $\ell \in [L]$.

3.4 Guarantees via dispersion

We now provide guarantees for the class of item-pricing mechanisms under unit-demand buyers, under which revenue is a particularly volatile function. In fact, the p-stability property does not hold for any non-trivial stability function p. We show that we can use another tool—called *dispersion* [5]—to obtain guarantees for learning with a noisy dataset. Therefore, we illustrate that p-stability is a sufficient but not necessary condition for obtaining robustness guarantees for a wide range of mechanism classes. We showcase this for item-pricing mechanisms.

Item-pricing mechanisms: Let there be n buyers and m items and as in Section 3.2, we assume there is only one unit of each item for sale. There is a non-anonymous price $r_{ij} \in \mathbb{R}$ for each buyer $i \in [n]$ and each item $j \in [m]$. First, buyer 1 arrives and buys the item $j \in [m]$ that maximizes his utility $v_{1j} - r_{1j}$ (or chooses not to buy if $v_{1j} < r_{1j}$ for all items $j \in [m]$). Next, buyer 2 arrives and selects the item among the remaining that maximizes his utility $v_{2j} - r_{2j}$ (or chooses not to buy). This process continues for each buyer $i \in [n]$. As in the previous section, we use the notation $\operatorname{rev}_{\boldsymbol{r}}(\boldsymbol{v})$ to denote the revenue of the mechanism with prices $\boldsymbol{r} \in \mathbb{R}^{nm}$ when the buyers have values $\boldsymbol{v} \in \mathbb{R}^{nm}$.

Under item-pricing mechanisms, we face an immediate obstacle which is that p-stability does not necessarily hold for any non-trivial choice of the function p: the empirically optimal prices over the noisy set $\mathcal{S}'$ might have terrible revenue on average over the true set $\mathcal{S}$ (Lemma B.11 in Appendix B.4).

The primary challenge in proving a stability guarantee is that an agent may have very different values for two items but very similar utilities given a set of prices $\boldsymbol{r}$. If we add a little noise to the buyer's values, their preference ordering over the items may flip. In turn, this may cause the buyer to select a low-cost item instead of a high-cost item, thus triggering a sharp drop in revenue. As a result, average revenue over a noisy training set may be completely different from average revenue over the uncorrupted training set. If, however, given any price vector $\boldsymbol{r}$, the buyers' utilities across items are not too similar (they are "dispersed"), their preferences will not change if a bit of noise is added to their values. Below, we formally capture this notion of dispersion.

Definition 3.12 ((ϵ, k)-dispersion)**.** Let $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\} \subset \mathbb{R}^{nm}_{\geq 0}$ be a set of valuation vectors. We say that $\mathcal{S}$ is (ϵ, k)-dispersed if for any price vector $\boldsymbol{r} \in \mathbb{R}^m$, there are at most k valuation vectors in $\mathcal{S}$ such that for some buyer $i \in [n]$, either:

1. For some item pair $j, j' \in [m]$, buyer i's utility for item j is within ϵ of her utility for item j', or
2. Buyer i's utility for some item j is between 0 and ϵ.

The parameter k allows for some slack: for some—but not all—of the valuation vectors in $\mathcal{S}$, the buyers' utilities can concentrate. Later in this section, we demonstrate that dispersion holds under mild assumptions. First, we provide a guarantee based on dispersion for optimizing prices using a noisy training set. The full proof is in Appendix B.4.

Theorem 3.13. *Let* $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\} \sim \mathcal{D}^L$ *be a set of* $(2\epsilon, k)$*-dispersed vectors. Let* $\mathcal{S}' = \left\{\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(L)}\right\} \subset \mathbb{R}^{nm}_{\geq 0}$ *be another set such that for all* $\ell \in [L]$, $\left\|\boldsymbol{v}^{(\ell)} - \boldsymbol{w}^{(\ell)}\right\|_\infty \leq \epsilon$. *Let* $\hat{\boldsymbol{r}}'$ *be empirically optimal over* $\mathcal{S}'$: $\hat{\boldsymbol{r}}' = \operatorname{argmax}_{\boldsymbol{r} \in \mathbb{R}^{nm}} \sum_{\ell=1}^{L} \operatorname{rev}_{\boldsymbol{r}}\left(\boldsymbol{w}^{(\ell)}\right)$. *With probability* $1 - \delta$ *over the draw of* $\mathcal{S}$, $\max_{\boldsymbol{r} \in \mathbb{R}^{nm}} \mathbb{E}_{\boldsymbol{v} \sim \mathcal{D}}\left[\operatorname{rev}_{\boldsymbol{r}}(\boldsymbol{v})\right] - \mathbb{E}\left[\operatorname{rev}_{\hat{\boldsymbol{r}}'}(\boldsymbol{v})\right] = O\left(\frac{k}{L} + \sqrt{\frac{1}{L}\left(nm\log(nm) + \log\frac{1}{\delta}\right)}\right)$.

Proof sketch. We first prove that for any price vector $\boldsymbol{r}$, average revenue over $\mathcal{S}$ is within $\frac{k}{L}$ of average revenue over $\mathcal{S}'$, for the following reason. Due to dispersion, for most of the valuation vectors

$\boldsymbol{v}^{(\ell)} \in \mathcal{S}$, the buyers will choose to buy the same set of goods regardless of whether their values are defined by $\boldsymbol{v}^{(\ell)}$ or $\boldsymbol{w}^{(\ell)}$, so the revenue remains constant. For at most k valuation vectors in $\mathcal{S}$, the allocation may change arbitrarily, but revenue is always bounded in the interval $[0, 1]$, which is why the bound contains the term $\frac{k}{L}$. Finally, to relate average and expected revenue, we use a generalization bound from prior research [39] which equals the second summand of our bound. $\square$

This result raises an important question: when will a set of valuations be dispersed? One example, also observed in prior research [5], is when the distribution over buyers' values is relatively "smooth" (*a la* smoothed analysis [43]), formalized as follows.

Definition 3.14. For any distribution over an abstract set $\mathcal{X}$ with probability density function $f : \mathcal{X} \to \mathbb{R}_{\geq 0}$, the density function is κ-*bounded* if $\max_{x \in \mathcal{X}} f(x) \leq \kappa$.

The proof of the following lemma is in Appendix B.4.

Lemma 3.15. *Suppose that for every buyer $i \in [n]$ and every pair of items $j, j' \in [m]$, buyer i's values for items j and j' have a κ-bounded joint density function. Then for any $\epsilon > 0$, with probability $1 - \delta$ over the draw $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(L)}\right\} \sim \mathcal{D}^L$, the set $\mathcal{S}$ is $(2\epsilon, k)$-dispersed with $k = 4Lnm^2\kappa\epsilon + O\left(nm^2\sqrt{L \log \frac{nm}{\delta}}\right)$.*

Theorem 3.13 and Lemma 3.15 imply learning guarantees for optimizing with noisy data when dispersion holds, even for these extremely volatile revenue functions.

4 Multi-task learning

We now show how our results from the previous section apply to multi-task mechanism design.

Setup: In this setting, there are T tasks, and each task $t \in [T]$ is defined by a distribution $\mathcal{D}^{(t)}$ over buyers' values. Each distribution could represent, for example, buyers from different regions or market segments. The learning algorithm receives a training set sampled from each task's distribution. Given a mechanism class parameterized by vectors $\boldsymbol{r} \in \mathbb{R}^d$ (which equal, for example, reserve prices) and the T training sets, our goal is to learn a parameter setting $\boldsymbol{r}^{(t)}$ for each task t with high expected revenue over $\mathcal{D}^{(t)}$. In particular, $\max_{\boldsymbol{r} \in \mathbb{R}^d} \mathbb{E}_{\boldsymbol{v} \sim \mathcal{D}^{(t)}}\left[\text{rev}_{\boldsymbol{r}}(\boldsymbol{v})\right] - \mathbb{E}_{\boldsymbol{v} \sim \mathcal{D}^{(t)}}\left[\text{rev}_{\boldsymbol{r}^{(t)}}(\boldsymbol{v})\right]$ should be small.

Task relatedness: If the distributions are completely unrelated, there is no hope that sharing information across tasks could improve learning. Thus, we require some notion of "task-relatedness". Under our model of task-relatedness, buyers' values across tasks are similar up to additive shifts which represent, for example, differences in their income brackets. More formally, for each buyer $i \in [n]$ and item $j \in [m]$, there is an underlying distribution $\mathcal{D}_{ij}$ over $\mathbb{R}_{\geq 0}$. Let $\mathcal{D} = \times_{i,j} \mathcal{D}_{ij}$ denote the cross product of these nm distributions. Each task $t \in [T]$ is defined by an unknown vector of *common-base values* $\boldsymbol{b}^{(t)} \in \mathbb{R}^n$. Intuitively, when $b_i^{(t)}$ is large, buyer i tends to be willing to pay more for items, and when $b_i^{(t)}$ is small, buyer i tends to be frugal. This model is inspired by the well-studied *common-base-value* model [9, 20]. For each task $t \in [T]$, our learning algorithm receives L samples $\boldsymbol{v}^{(1,t)}, \ldots, \boldsymbol{v}^{(L,t)} \in \mathbb{R}^{nm}$. Each sample $\boldsymbol{v}^{(\ell,t)}$ is generated by sampling $\boldsymbol{z}^{(\ell,t)} \sim \mathcal{D}$ and defining buyer i's value for item j to be $v_{ij}^{(\ell,t)} = b_i^{(t)} + z_{ij}^{(\ell,t)}$. This defines a distribution $\mathcal{D}_{ij}^{(t)}$ over buyer i's value for item j. We use the notation $\mathcal{D}^{(t)} = \times_{i,j} \mathcal{D}_{ij}^{(t)}$ to denote the cross product of these nm distributions for task t. The learning algorithm does not know the vectors $\boldsymbol{b}^{(t)}$ or $\boldsymbol{z}^{(\ell,t)}$; it only observes the valuation vectors $\boldsymbol{v}^{(\ell,t)}$. Throughout this section, we use the (slight abuse of) notation $\boldsymbol{b}^{(t)} + \boldsymbol{z}^{(\ell,t)}$ to denote $\boldsymbol{v}^{(\ell,t)}$. In Appendix D, we supply empirical evidence that demonstrates that this is a reasonable model of task-relatedness on real-world auction data.

Sample bootstrapping: Our multi-task approach follows from the observation that for each task $t \in [T]$, $\mathcal{S}_t = \left\{\boldsymbol{b}^{(t)} + \boldsymbol{z}^{(\ell,\tau)} : \tau \in [T], \ell \in [L]\right\}$ is a training set of LT samples from the tth task's distribution $\mathcal{D}^{(t)}$ since each vector $\boldsymbol{z}^{(\ell,\tau)}$ is sampled from $\mathcal{D}$. However, we cannot compile this training set since we do not know the base values $\boldsymbol{b}^{(t)}$. Nonetheless, in Appendix C, we show that it is straightforward to compute a set of vectors $\mathcal{S}'_t = \left\{\boldsymbol{w}^{(\ell,t,\tau)} : \ell \in [L], \tau \in [T]\right\}$ that closely approximate* the vectors in $\mathcal{S}_t$: with high probability, for all pairs of tasks $t, \tau \in [T]$ and indices

*In some cases, we show that it is also possible to exactly recover the vectors in $\mathcal{S}_t$ (Appendix C.2).

$\ell \in [L]$, $\left\|\boldsymbol{b}^{(t)} + \boldsymbol{z}^{(\ell,\tau)} - \boldsymbol{w}^{(\ell,t,\tau)}\right\|_\infty = \tilde{O}\left(\sqrt{\frac{1}{Lm}}\right)$. From our results in Section 3, we know that it is possible to learn a mechanism with high expected revenue over the distribution $\mathcal{D}^{(t)}$ using the noisy training set $\mathcal{S}'_t$. We summarize the resulting guarantees below. All proofs are in Appendix C.

Multi-item non-anonymous second-price auctions. Our results from Section 3.2 imply the following guarantee for learning a high-revenue multi-item second-price auction with non-anonymous reserves under additive bidders.

Theorem 4.1. *For each task $t \in [T]$, let $\hat{\boldsymbol{r}}'_t$ be the empirically optimal reserve vector over the set $\mathcal{S}'_t$: $\hat{\boldsymbol{r}}'_t = \operatorname{argmax}_{\boldsymbol{r}\in\mathbb{R}^{nm}} \sum_{\ell=1}^{L}\sum_{\tau=1}^{T} \operatorname{rev}_{\boldsymbol{r}}\left(\boldsymbol{w}^{(\ell,t,\tau)}\right)$. With probability $1-\delta$, for every task $t \in [T]$, $\max_{\boldsymbol{r}\in\mathbb{R}^{nm}} \mathbb{E}_{\boldsymbol{v}\sim\mathcal{D}^{(t)}}\left[\operatorname{rev}_{\boldsymbol{r}}(\boldsymbol{v}) - \operatorname{rev}_{\hat{\boldsymbol{r}}'_t}(\boldsymbol{v})\right] = \tilde{O}\left(\sqrt{\frac{1}{LT}\left(nm + \log\frac{T}{\delta}\right) + \frac{m}{L}\log\frac{nT}{\delta}}\right)$.*

This theorem implies that for any $\gamma \in (0,1)$, when $L = \tilde{\Omega}\left(m/\gamma^2\right)$ and $T = \tilde{\Omega}(n)$, the revenue of the auction defined by the reserves $\hat{\boldsymbol{r}}'_t$ is within γ of optimal: $\max \mathbb{E}_{\boldsymbol{v}\sim\mathcal{D}^{(t)}}\left[\operatorname{rev}_{\boldsymbol{r}}(\boldsymbol{v})\right] - \mathbb{E}_{\boldsymbol{v}\sim\mathcal{D}^{(t)}}\left[\operatorname{rev}_{\hat{\boldsymbol{r}}'_t}(\boldsymbol{v})\right] \le \gamma$. In contrast, the best-known single-task sample complexity guarantee requires $L = \tilde{\Omega}\left(nm/\gamma^2\right)$ [39][†]. Our per-task sample complexity is smaller by a multiplicative factor equalling the number of buyers.

Lotteries. Our results from Section 3.3 imply that under unit-demand buyers, optimizing lottery parameters using the noisy training sets $\mathcal{S}'_t$ results in nearly optimal revenue.

Theorem 4.2. *For each task t, let $\hat{\boldsymbol{r}}'_t$ be the empirically optimal lottery parameter vector over the set $\mathcal{S}'_t$: $\hat{\boldsymbol{r}}'_t = \operatorname{argmax}_{\boldsymbol{r}\in\mathbb{R}^{n(m+1)}} \sum_{\ell=1}^{L}\sum_{\tau=1}^{T} \operatorname{rev}_{\boldsymbol{r}}\left(\boldsymbol{w}^{(\ell,t,\tau)}\right)$. With probability $1-\delta$, for every $t \in [T]$, $\max_{\boldsymbol{r}\in\mathbb{R}^{n(m+1)}} \mathbb{E}_{\boldsymbol{v}\sim\mathcal{D}^{(t)}}\left[\operatorname{rev}_{\boldsymbol{r}}(\boldsymbol{v}) - \operatorname{rev}_{\hat{\boldsymbol{r}}'_t}(\boldsymbol{v})\right] = \tilde{O}\left(\sqrt{\frac{1}{LT}\left(nm + \log\frac{T}{\delta}\right) + \frac{n^2}{Lm}\log\frac{nT}{\delta}}\right)$.*

Theorem 4.2 implies that for any $\gamma \in (0,1)$, when $L = \tilde{\Omega}\left(n/\gamma^2\right)$, $T = \tilde{\Omega}(m)$, and there are more items than buyers ($m > n$), the revenue of the lottery defined by $\hat{\boldsymbol{r}}'_t$ is within γ of optimal: $\max \mathbb{E}_{\boldsymbol{v}\sim\mathcal{D}^{(t)}}\left[\operatorname{rev}_{\boldsymbol{r}}(\boldsymbol{v})\right] - \mathbb{E}_{\boldsymbol{v}\sim\mathcal{D}^{(t)}}\left[\operatorname{rev}_{\hat{\boldsymbol{r}}'_t}(\boldsymbol{v})\right] \le \gamma$. Meanwhile, the best-known single-task sample complexity bound requires $L = \tilde{\Omega}\left(nm/\gamma^2\right)$ [6]. Our approach's per-task sample complexity is better by a multiplicative factor of m.

5 Conclusions

We provided guarantees for learning high-revenue mechanisms with noisy data. Learning in the presence of noise is particularly challenging in mechanism design because revenue functions are volatile, exhibiting many jump discontinuities. We were nonetheless able to provide revenue guarantees only under the assumption that the magnitude of the noise is bounded. Our guarantees apply to the dominant approach to learning-based mechanism design: empirical revenue maximization (ERM). We demonstrated the application of our guarantees to multi-task mechanism design. We thus initiated the study of both learning with noisy data and multi-task learning in mechanism design.

Future directions: (1) We focused on learning with bounded noise rather than noise of arbitrary magnitude (the full adversarial game). This is an exciting future direction if any guarantees are indeed possible when bids may be arbitrarily altered. This may require the development of new learning algorithms beyond ERM. (2) A major bottleneck for empirical evaluations in learning-based mechanism design is the lack of public datasets—we are only aware of one, from eBay [37]. We focused on theoretical guarantees, but applied research on robust and multi-task learning in mechanism design is a great future direction. (3) Building on our results for multi-task learning, what other notions of task-relatedness permit strong multi-task sample complexity bounds?

Negative societal impacts: This paper falls into the broader line of research on using machine learning for mechanism design. In this direction, collecting individuals' data and using it to fine-tune prices could have negative privacy implications. The development of privacy-preserving approaches to mechanism design via machine learning is a great direction for future research.

[†]The pseudo-dimension of the mechanism class is at least nm [6], which suggests that the dependence on n may be necessary in the single-task setting.

References

[1] Noga Alon, Moshe Babaioff, Yannai A Gonczarowski, Yishay Mansour, Shay Moran, and Amir Yehudayoff. Submultiplicative Glivenko-Cantelli and uniform convergence of revenues. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[2] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853, 2005.

[3] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 167–190, 2015.

[4] Maria-Florina Balcan, Avrim Blum, Jason Hartline, and Yishay Mansour. Reducing mechanism design to algorithm design via machine learning. *Journal of Computer and System Sciences*, 74: 78–89, December 2008.

[5] Maria-Florina Balcan, Travis Dick, and Ellen Vitercik. Dispersion for data-driven algorithm design, online learning, and private optimization. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2018.

[6] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. A general theory of sample complexity for multi-item profit maximization. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2018.

[7] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. Estimating approximate incentive compatibility. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2019.

[8] Maria-Florina Balcan, Travis Dick, and Wesley Pegden. Semi-bandit optimization in the dispersed setting. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

[9] MohammadHossein Bateni, Sina Dehghani, MohammadTaghi Hajiaghayi, and Saeed Seddighin. Revenue maximization for selling multiple correlated items. In *Proceedings of the European Symposium on Algorithms (ESA)*, pages 95–105. Springer, 2015.

[10] Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000.

[11] Xiaohui Bei and Zhiyi Huang. Bayesian incentive compatibility via fractional assignments. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 720–733. SIAM, 2011.

[12] Shai Ben-David and Reba Schuller Borbely. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine learning*, 73(3):273–287, 2008.

[13] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79 (1-2):151–175, 2010.

[14] Kush Bhatia, Prateek Jain, and Purushottam Kar. Robust regression via hard thresholding. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 721–729, 2015.

[15] Avrim Blum, Nika Haghtalab, Ariel D Procaccia, and Mingda Qiao. Collaborative pac learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 2392–2401, 2017.

[16] Patrick Briest, Shuchi Chawla, Robert Kleinberg, and S Matthew Weinberg. Pricing randomized allocations. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 585–597. SIAM, 2010.

[17] Sébastien Bubeck, Nikhil R Devanur, Zhiyi Huang, and Rad Niazadeh. Online auctions and multi-scale online learning. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2017.

[18] Yang Cai and Constantinos Daskalakis. Learning multi-item auctions with (or without) samples. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2017.

[19] Yang Cai, Argyris Oikonomou, Grigoris Velegkas, and Mingfei Zhao. An efficient epsilon-bic to bic transformation and its application to black-box reduction in revenue maximization. *arXiv preprint arXiv:1911.10172*, 2019.

[20] Shuchi Chawla, David Malec, and Balasubramanian Sivan. The power of randomness in Bayesian optimal mechanism design. *Games and Economic Behavior*, 91:297–317, 2015.

[21] Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 2014.

[22] Nikhil R Devanur, Zhiyi Huang, and Christos-Alexandros Psomas. The sample complexity of auctions with side information. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 2016.

[23] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019.

[24] Edith Elkind. Designing and learning optimal finite support auctions. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

[25] Tomer Galanti, Lior Wolf, and Tamir Hazan. A theoretical framework for deep transfer learning. *Information and Inference: A Journal of the IMA*, 5(2):159–209, 2016.

[26] Yannai A Gonczarowski and Noam Nisan. Efficient empirical revenue maximization in single-parameter auction environments. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 856–868, 2017.

[27] Yannai A Gonczarowski and S Matthew Weinberg. The sample complexity of up-to-ε multi-dimensional revenue maximization. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2018.

[28] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[29] Chenghao Guo, Zhiyi Huang, and Xinzhi Zhang. Settling the sample complexity of single-parameter revenue maximization. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 2019.

[30] Rishi Gupta and Tim Roughgarden. A PAC approach to application-specific algorithm selection. *SIAM Journal on Computing*, 46(3):992–1017, 2017.

[31] Jason D Hartline and Brendan Lucier. Bayesian algorithmic mechanism design. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 301–310, 2010.

[32] Zhiyi Huang, Yishay Mansour, and Tim Roughgarden. Making the most of your samples. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2015.

[33] Anton Likhodedov and Tuomas Sandholm. Methods for boosting revenue in combinatorial auctions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2004.

[34] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[35] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Proceedings of the Conference on Learning Theory (COLT)*, 2009.

[36] Daniel McNamara and Maria-Florina Balcan. Risk bounds for transferring representations with and without fine-tuning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2373–2381, 2017.

[37] Mehryar Mohri and Andrés Muñoz Medina. Learning theory and algorithms for revenue optimization in second price auctions with reserve. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014.

[38] Jamie Morgenstern and Tim Roughgarden. On the pseudo-dimension of nearly optimal auctions. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

[39] Jamie Morgenstern and Tim Roughgarden. Learning simple auctions. In *Proceedings of the Conference on Learning Theory (COLT)*, 2016.

[40] Andrés Muñoz Medina and Sergei Vassilvitskii. Revenue optimization with approximate bid predictions. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[41] Tim Roughgarden and Okke Schrijvers. Ironing in the dark. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2016.

[42] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *arXiv preprint arXiv:1804.11285*, 2018.

[43] Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3):385–463, 2004.

[44] Vasilis Syrgkanis. A sample complexity measure with applications to learning optimal auctions. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[45] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[46] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

6 Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We discussed the limitations of our work in the "Future directions" paragraph of Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We discussed any potential negative societal impacts in Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] We provide full, detailed proofs of all results. Due to space constraints, the main body includes proof sketches of some of the results, and the full proofs of these results are in the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]