Distribution Free Profit Maximization via Online Auctions

Aaron Roth

University of Pennsylvania

April 8 2025

Overview

- Suppose we want to maximize revenue in a digital goods setting but with pricings rather than auctions?

Overview

- Suppose we want to maximize revenue in a digital goods setting but with pricings rather than auctions?
- Remember it can be hard to run auctions... We need all bidders there at the same time!

Overview

- Suppose we want to maximize revenue in a digital goods setting but with pricings rather than auctions?
- Remember it can be hard to run auctions... We need all bidders there at the same time!
- Bidders arriving online don't necessarily have their valuations drawn from a distribution. (Can be chosen by an adaptive adversary)

Overview

- Suppose we want to maximize revenue in a digital goods setting but with pricings rather than auctions?
- Remember it can be hard to run auctions... We need all bidders there at the same time!
- Bidders arriving online don't necessarily have their valuations drawn from a distribution. (Can be chosen by an adaptive adversary)
- We'll solve this by bringing the class full circle — using the polynomial weights algorithm!

Review

- Recall our solution from last lecture: The Random Sampling Auction.

Review

- Recall our solution from last lecture: The Random Sampling Auction.
- Randomly partition bidders into to buckets, compute the optimal revenue in each bucket, and use that estimate in the other bucket.

Review

- Recall our solution from last lecture: The Random Sampling Auction.
- Randomly partition bidders into to buckets, compute the optimal revenue in each bucket, and use that estimate in the other bucket.
- i.e. solve a *statistical estimation/learning* problem to maximize revenue.

Review

- Recall our solution from last lecture: The Random Sampling Auction.
- Randomly partition bidders into to buckets, compute the optimal revenue in each bucket, and use that estimate in the other bucket.
- i.e. solve a *statistical estimation*/*learning* problem to maximize revenue.
- Can we do something similar without having all bidders there up front? An *online* learning problem?

This Lecture

- Goal: find a truthful *online* auction to approximate the optimal revenue.

This Lecture

- Goal: find a truthful *online* auction to approximate the optimal revenue.
  - Bidders sequentially report their valuations and then receive an allocation before the next bidder arrives.

This Lecture

- Goal: find a truthful *online* auction to approximate the optimal revenue.
  - Bidders sequentially report their valuations and then receive an allocation before the next bidder arrives.
  - Similar ideas would work to give a dynamic pricing scheme, but an online auction is a little easier. We'll see a pricing scheme next lecture.

This Lecture

- Goal: find a truthful *online* auction to approximate the optimal revenue.
  - Bidders sequentially report their valuations and then receive an allocation before the next bidder arrives.
  - Similar ideas would work to give a dynamic pricing scheme, but an online auction is a little easier. We'll see a pricing scheme next lecture.
- Recall our revenue benchmark: $\mathrm{OPT}^{\geq k}(v) = \max_{j \geq k}(j \cdot v_{(j)})$. The random sampling auction achieved a 4 approximation to $\mathrm{OPT}^{\geq 2}(v)$

This Lecture

- Goal: find a truthful *online* auction to approximate the optimal revenue.
  - Bidders sequentially report their valuations and then receive an allocation before the next bidder arrives.
  - Similar ideas would work to give a dynamic pricing scheme, but an online auction is a little easier. We'll see a pricing scheme next lecture.
- Recall our revenue benchmark: $\mathrm{OPT}^{\geq k}(v) = \max_{j \geq k}(j \cdot v_{(j)})$. The random sampling auction achieved a 4 approximation to $\mathrm{OPT}^{\geq 2}(v)$
- Well aim for a $1 + \epsilon$ approximation for larger k.

Our Setting

Definition

In an online digital goods auction, we have n bidders with valuations $v_i \in [0, 1]$.

Our Setting

Definition

In an online digital goods auction, we have n bidders with valuations $v_i \in [0, 1]$.

- At time t, bidder t arrives and reports valuation v_t'.

Our Setting

Definition

In an online digital goods auction, we have n bidders with valuations $v_i \in [0, 1]$.

- At time t, bidder t arrives and reports valuation v'_t.
- An item is allocated according to rule $x_t(v'_1, \ldots, v'_t)$, and payment $p_t(v'_1, \ldots, v'_t)$ is collected. Note that the allocation and payment rule is allowed to depend on *previous* bidders, but not *future* bidders.

Auction Format

It will be helpful for us to think about a particularly simple kind of allocation and payment rule:

Definition

In a *take-it-or-leave-it* (TIOLI) auction:

Auction Format

It will be helpful for us to think about a particularly simple kind of allocation and payment rule:

Definition

In a *take-it-or-leave-it* (TIOLI) auction:

- At time t, a fixed price $s_t = s_t(v'_1, \ldots, v'_{t-1})$ is computed.

Auction Format

It will be helpful for us to think about a particularly simple kind of allocation and payment rule:

Definition

In a *take-it-or-leave-it* (TIOLI) auction:

- At time t, a fixed price $s_t = s_t(v'_1, \ldots, v'_{t-1})$ is computed.
- The item is sold according to the following allocation and payment rules:

$$x_t(v'_1, \ldots, v_{t-1}, v'_t) = 1 \Leftrightarrow v'_t \geq s_t \qquad p_t(v'_1, \ldots, v'_{t-1}) = s_t$$

Auction Format

It will be helpful for us to think about a particularly simple kind of allocation and payment rule:

Definition

In a *take-it-or-leave-it* (TIOLI) auction:

- At time t, a fixed price $s_t = s_t(v'_1, \ldots, v'_{t-1})$ is computed.
- The item is sold according to the following allocation and payment rules:

$$x_t(v'_1, \ldots, v_{t-1}, v'_t) = 1 \Leftrightarrow v'_t \geq s_t \qquad p_t(v'_1, \ldots, v'_{t-1}) = s_t$$

i.e. the item is sold at a fixed price s_t to bidders with valuation above the price, and the price s_t is computed *independently* of bidder t's own bid.

Auction Format

A simple observation:

Theorem

Any take-it-or-leave-it auction is dominant strategy truthful.

Auction Format

A simple observation:

Theorem

Any take-it-or-leave-it auction is dominant strategy truthful.

Proof.

Since the price that bidder t faces is computed independently of his own bid, over/under-reporting does not influence the price – it can only result in agent t winning the item at a price he was not willing to pay, or failing to win the item even when he would have been willing to pay the price. $\square$

Auction Format

A simple observation:

Theorem

Any take-it-or-leave-it auction is dominant strategy truthful.

Proof.

Since the price that bidder t faces is computed independently of his own bid, over/under-reporting does not influence the price – it can only result in agent t winning the item at a price he was not willing to pay, or failing to win the item even when he would have been willing to pay the price. □

Its not hard to see that it is without loss of generality to consider TIOLI auctions... In single parameter domains, truthful auctions must be monotone. For deterministic auctions, this means that the allocation rule for each bidder must be determined by a fixed, bid-independent threshold (i.e. the fixed price)).

Remembering Polynomial Weights

- Our goal: *learn* the best fixed price.

Remembering Polynomial Weights

- Our goal: *learn* the best fixed price.
- The idea: Use the polynomial weights algorithm, using prices as experts.

Remembering Polynomial Weights

- Our goal: *learn* the best fixed price.
- The idea: Use the polynomial weights algorithm, using prices as experts.
- Recall the setting and guarantees of the polynomial weights algorithm:

Remembering Polynomial Weights

- Our goal: *learn* the best fixed price.
- The idea: Use the polynomial weights algorithm, using prices as experts.
- Recall the setting and guarantees of the polynomial weights algorithm:
  - Given a collection of N experts, each of whom experience gains $g_i^t \in [0, 1]$ each day t.

Remembering Polynomial Weights

- Our goal: *learn* the best fixed price.
- The idea: Use the polynomial weights algorithm, using prices as experts.
- Recall the setting and guarantees of the polynomial weights algorithm:
  - Given a collection of N experts, each of whom experience gains $g_i^t \in [0, 1]$ each day t.
  - The polynomial weights algorithm selects an expert each day and experiences its gain.

Remembering Polynomial Weights

- Our goal: *learn* the best fixed price.
- The idea: Use the polynomial weights algorithm, using prices as experts.
- Recall the setting and guarantees of the polynomial weights algorithm:
  - Given a collection of N experts, each of whom experience gains $g_i^t \in [0, 1]$ each day t.
  - The polynomial weights algorithm selects an expert each day and experiences its gain.
  - Guarantees that after T rounds: with update parameter ϵ is able to select experts so as to achieve expected gain after T rounds:
$$G_{PW}^T \geq \max_{k \in [N]} G_k^T - 2\sqrt{T \ln(N)}$$

Using Polynomial Weights

- Lets fix some collection of N prices $N \subseteq [0, 1]$ and treat them as "experts".

Using Polynomial Weights

- Lets fix some collection of N prices $N \subseteq [0, 1]$ and treat them as "experts".
- What should their gains be?

Using Polynomial Weights

- Lets fix some collection of N prices $N \subseteq [0, 1]$ and treat them as "experts".
- What should their gains be?
- If we use price s on bidder t, we obtain revenue:

$$r_s^t = \begin{cases} s, & \text{if } v_t \geq s; \\ 0, & \text{if } v_t < s. \end{cases}$$

Using Polynomial Weights

- Lets fix some collection of N prices $N \subseteq [0,1]$ and treat them as "experts".
- What should their gains be?
- If we use price s on bidder t, we obtain revenue:

$$r_s^t = \begin{cases} s, & \text{if } v_t \geq s; \\ 0, & \text{if } v_t < s. \end{cases}$$

- So these are our gains. $g_s^t = r_s^t$.

Using Polynomial Weights

- Let Rev_p^T denote the revenue of using fixed price p for the first T bidders:

$$Rev_p^T = p \cdot |\{i \leq T : v_i \geq p\}|$$

Using Polynomial Weights

- Let Rev_p^T denote the revenue of using fixed price p for the first T bidders:

$$Rev_p^T = p \cdot |\{i \leq T : v_i \geq p\}|$$

- By construction, this is the same as the cumulative gain of an expert corresponding to p: $G_p^T = Rev_p^T$.

Using Polynomial Weights

- Let Rev_p^T denote the revenue of using fixed price p for the first T bidders:

$$Rev_p^T = p \cdot |\{i \leq T : v_i \geq p\}|$$

- By construction, this is the same as the cumulative gain of an expert corresponding to p: $G_p^T = Rev_p^T$.
- If we use the PW to select a price from some set N at every round, we get a Take-It-Or-Leave-It mechanism, which is dominant strategy truthful. Moreover, we are guaranteed:

$$Rev_{PW}^T \geq \max_{p \in N} Rev_p^T - 2\sqrt{T \ln(N)}$$

Using Polynomial Weights

- Let Rev_p^T denote the revenue of using fixed price p for the first T bidders:

$$Rev_p^T = p \cdot |\{i \leq T : v_i \geq p\}|$$

- By construction, this is the same as the cumulative gain of an expert corresponding to p: $G_p^T = Rev_p^T$.
- If we use the PW to select a price from some set N at every round, we get a Take-It-Or-Leave-It mechanism, which is dominant strategy truthful. Moreover, we are guaranteed:

$$Rev_{PW}^T \geq \max_{p \in N} Rev_p^T - 2\sqrt{T \ln(N)}$$

- So how should we choose our set of prices N?

Using Polynomial Weights

- Let Rev_p^T denote the revenue of using fixed price p for the first T bidders:

$$Rev_p^T = p \cdot |\{i \leq T : v_i \geq p\}|$$

- By construction, this is the same as the cumulative gain of an expert corresponding to p: $G_p^T = Rev_p^T$.
- If we use the PW to select a price from some set N at every round, we get a Take-It-Or-Leave-It mechanism, which is dominant strategy truthful. Moreover, we are guaranteed:

$$Rev_{PW}^T \geq \max_{p \in N} Rev_p^T - 2\sqrt{T \ln(N)}$$

- So how should we choose our set of prices N?
- There is a tradeoff – choosing a larger set makes $\max_{p \in N} Rev_p^T$ closer to $\mathrm{OPT}(v)$, but also makes $\ln(N)$ larger...

Choosing the Experts

- Consider choosing prices that are multiples of some $\alpha > 0$:

$$N = \{\alpha, 2\alpha, 3\alpha, \ldots, 1\}$$

.

Choosing the Experts

- Consider choosing prices that are multiples of some $\alpha > 0$:

$$N = \{\alpha, 2\alpha, 3\alpha, \dots, 1\}$$

.

- We have that $|N| = \frac{1}{\alpha}$.

Choosing the Experts

- Consider choosing prices that are multiples of some $\alpha > 0$:
$$N = \{\alpha, 2\alpha, 3\alpha, \ldots, 1\}$$
.
- We have that $|N| = \frac{1}{\alpha}$.
- We also know that:
$$\max_{p \in N} Rev_p^T \geq \max_{p \in [0,1]} Rev_p^T - \alpha \cdot n$$
Because for every $p \in [0,1]$ there is a $p' \in N$ such that $p - \alpha \leq p' \leq p$.

Choosing the Experts

- Combining these guarantees we get:

$$Rev^n_{PW} \geq \max_{p \in [0,1]} Rev^n_p - 2\sqrt{n \ln(\frac{1}{\alpha})} - \alpha n$$

Choosing the Experts

- Combining these guarantees we get:

$$Rev^n_{PW} \geq \max_{p\in[0,1]} Rev^n_p - 2\sqrt{n\ln(\frac{1}{\alpha})} - \alpha n$$

- Choosing α to be $1/n$ we get:

$$Rev^n_{PW} \geq \max_{p\in[0,1]} Rev^n_p - 3\sqrt{n\ln(n)}$$

Interpreting the Guarantee

$$Rev_{PW}^{n} \geq \mathrm{OPT} - 3\sqrt{n \ln(n)}$$

- Strictly speaking, this guarantee is incomparable to the 4-approximation we derived last time (because it is additive).

Interpreting the Guarantee

$$Rev_{PW}^{n} \geq \mathrm{OPT} - 3\sqrt{n \ln(n)}$$

- Strictly speaking, this guarantee is incomparable to the 4-approximation we derived last time (because it is additive).
- But we would naturally expect that optimal revenue should grow with n.

Interpreting the Guarantee

$$Rev_{PW}^{n} \geq \mathrm{OPT} - 3\sqrt{n\ln(n)}$$

- Strictly speaking, this guarantee is incomparable to the 4-approximation we derived last time (because it is additive).
- But we would naturally expect that optimal revenue should grow with n.
- This gives us (asymptotically) a 1-approximation to OPT whenever OPT grows as $\mathrm{OPT}(n) \in \omega(\sqrt{n\log n})$.

Interpreting the Guarantee

$$Rev_{PW}^{n} \geq \mathrm{OPT} - 3\sqrt{n \ln(n)}$$

- Strictly speaking, this guarantee is incomparable to the 4-approximation we derived last time (because it is additive).
- But we would naturally expect that optimal revenue should grow with n.
- This gives us (asymptotically) a 1-approximation to OPT whenever OPT grows as $\mathrm{OPT}(n) \in \omega(\sqrt{n \log n})$.
- e.g. it suffices if with constant probability bidders have valuations $v_i \geq \log n/\sqrt{n}$.

Interpreting the Guarantee

$$Rev^n_{PW} \geq \mathrm{OPT} - 3\sqrt{n\ln(n)}$$

- Strictly speaking, this guarantee is incomparable to the 4-approximation we derived last time (because it is additive).
- But we would naturally expect that optimal revenue should grow with n.
- This gives us (asymptotically) a 1-approximation to OPT whenever OPT grows as $\mathrm{OPT}(n) \in \omega(\sqrt{n\log n})$.
- e.g. it suffices if with constant probability bidders have valuations $v_i \geq \log n/\sqrt{n}$.
- True for *any* fixed nontrivial distribution as $n \rightarrow \infty$.

Thanks!

See you next class — stay healthy!