Online Learning w/ Multiple Reserves

- We saw how to learn good mechanisms from IID data samples.
- What if bids are not IID but follow some stochastic process?
- What if we want to be robust to any such process?
- Can we do "online learning" for mechanisms?

* New Setting:
  - Every day learner picks a mechanism $M^t$
  - Adversary picks a value vector $v^+ = (v^+_1, \ldots, v^+_T)$
  - Learner receives reward equal to revenue of mechanism on $v^+$: $\text{rev}(M^t, v^+)$
  - Goal: Compete w/ best fixed mechanism in some class, i.e. $\text{Regret}$.

$$\text{Regret} = \mathbb{E}\left[\max_{M \in \mathcal{M}} \sum_{t=1}^{T} \text{rev}(M, v^+) - \sum_{t=1}^{T} \text{rev}(M^t, v^+)\right]$$

$$= o(T)$$
- Strictly harder than the PAC learning of mechanisms that we saw.

- **Today:** \( \mathcal{M} = \{ \text{second-price w/ player specific reserves} \} \)
  - Start from easy: \( \mathcal{M} = \{ \text{second-price w/ single reserve} \} \)
  - Move to harder \( \mathcal{M} \): achieve \( \frac{1}{2} - \text{Regret} : \)
    \[
    \mathbb{E} \left[ \frac{1}{2} \max_{M} \sum_{t=1}^{T} \text{rev}(M_t, v^t) - \sum_{t=1}^{T} \text{rev}(M^*, v^t) \right] = o(1) \]
    in poly-time.

- **Single Reserve:**
  - Discretize reserve: \( r \in \{ \varepsilon, 2\varepsilon, \ldots, 1 \} \subseteq [0, 1] \)
    \[
    \max_{r \in [0, 1]} \sum_{t=1}^{T} \text{rev}(r, v^t) = \max_{r \in [0, 1]} \sum_{t=1}^{T} \text{rev}(r, v^t) - \varepsilon T \geq \sum_{t=1}^{T} \text{rev}(r, v^t) - \varepsilon T \]
  - Whatever \( r \in [0, 1] \) you give me, round down to closest multiple of \( \varepsilon \).

  - Now we have \( \frac{1}{\varepsilon} \) possible reserves.

  - We are back to the expert setting:
    - \( \frac{1}{\varepsilon} \) actions, each action has reward \( \text{rev}(r, v^t) \)

  - Apply your favorite no-regret algorithm, from the first part of the course, to get: \( (\text{e.g. Exponential Weight Update}) \)
    \[
    r_t \in \left( \sqrt{\log \frac{1}{\varepsilon}} \right) \]

New Section 1 Page 2
\[ \mathbb{E} \left[ \max_{\nu \in [0,1]} \sum_{t=1}^{T} \text{rev}(r_t, \nu^*_{t}) - \sum_{t=1}^{T} \text{rev}(r_t, \nu^*_{t}) \right] \leq C \sqrt{\frac{\log \frac{1}{\varepsilon}}{T}} \]

\[ \Rightarrow \text{Regret} \leq C \sqrt{\frac{\log \frac{1}{\varepsilon}}{T}} + \varepsilon T \leq O\left(\sqrt{T \log T}\right) \]

**Multiple Reserves**

- Even if we discretize each reserve at a grid of \( \varepsilon \), we have \( \left(\frac{1}{\varepsilon}\right)^n \) reserves.

- Computationally inefficient: e.g. in exponential weights need to keep track of \( \left(\frac{1}{\varepsilon}\right)^n \) weights.

- Even though: \( \text{regret} \leq \sqrt{\frac{\log \left(\frac{1}{\varepsilon}\right)^n}{T}} = \sqrt{n \log \frac{1}{\varepsilon}} \)

**Idea:**
- Find an upper bound on the revenue of any auction in the class.

  - Show that a constant factor of this upper bound is achievable by an algorithm.

  - Show that no-regret on the upper bound decomposes on a set of \( n \) separable problems, so we can essentially run a no-regret alg for each reserve separately.

**Lemma**
For any value vector \( \nu \),

\[ \text{rev}(r, \nu) \leq \nu_{(2)} + (r_{(1)} - \nu_{(2)}) \mathbb{1}_{\{r_{(1)} \in [\nu_{(2)}, \nu_{(1)}]\}} \]
where (2) is the second highest value bidder and
(1) is the highest value bidder.

If \( r \) s that the highest value bidder
doesn't get the item.

Then revenue in that case is at most second
highest value.

If highest bidder does get the item then revenue
is \( \max \sum v_{(2)}, r_{(1)} \).

So \( \text{rev} (v, r) \leq \max \sum v_{(2)}, r_{(1)} 1 \{ v_{(1)} \geq r_{(1)} \} \)

\( \leq v_{(2)} + (r_{(1)} - v_{(2)}) 1 \{ r_{(1)} \in [v_{(2)}, v_{(1)}] \} \).

• Let \( x_i (v) = 1 \sum i \) is highest value bidder \( 3 \Rightarrow \)

\( \text{rev} (r, v) \leq v_{(2)} + \sum_i (r_i - v_{(2)}) 1 \sum v_{(2)} \in [v_{(2)}, v_{(i)}] \} x_i (v) \)

\( q_i (r_i, v) \)

This is a function only of \( r_i \).

• Suppose that we run a separate online learning
algorithm for each \( q_i (r_i, v^+) \), i.e.

\( \mathbb{E} \left[ \max_{r_i \in [0, 1]} \sum q_i (r_i, v^+) - \sum q_i (r_i^+, v^+) \right] \leq \sqrt{\frac{\log \frac{1}{\delta}}{T}} \)

• Then
Then
\[
\mathbb{E}\left[ \max_{\mathbf{r}^* \in (\mathcal{Q})_\mathcal{R}^T} \sum_{t} \sum_{i} q_i(r_i, v^t) - \sum_{t} \sum_{i} q_i(r_i^+, v^t) \right] \leq n \sqrt{\frac{\log \frac{1}{\delta}}{T}}
\]

This maximization is separable across i

Now specify at each iteration we pick \( r^+ \)
with prob. \( \frac{1}{2} \) and 0 w.p. \( \frac{1}{2} \).

\[
\mathbb{E}\left[ \text{rev}(R, v^+)^T \right] = \frac{1}{2} V^+_{(2)} + \frac{1}{2} \text{rev}(\mathbf{r}^+)^T, v^+)
\]
\[
\geq \frac{1}{2} V^+_{(2)} + \frac{1}{2} \left( r^+_{(1)} - v^+_{(2)} \right) \sum_{i \in \mathcal{I}} \bar{\phi}_i(r^+_{(2)}, v^+)
\]
\[
\geq \frac{1}{2} V^+_{(2)} + \frac{1}{2} \sum_{i} q_i(r^+_{(2)}, v^+)
\]

By no-regret: if \( \mathbf{p^*} \) is optimal reserve price vector
in \( [0, 1]^n \)
\[
\mathbb{E}\left[ \text{rev}(R, v^+) \right] \geq \frac{1}{2} \sum_{t} V^+_{(2)} + \frac{1}{2} \sum_{i} q_i(\mathbf{p^*}, v^+) - n \sqrt{\frac{\log \frac{1}{\delta}}{T}}
\]
\[
\geq \frac{1}{2} \sum_{t} \left( V^+_{(2)} + \sum_{i} q_i(\mathbf{p^*}, v^+) \right) - n \sqrt{\frac{\log \frac{1}{\delta}}{T}}
\]
\[
\geq \frac{1}{2} \sum_{t} \text{rev}(\mathbf{p^*}, v^+) - n \sqrt{\frac{\log \frac{1}{\delta}}{T}}
\]

We also lose an extra \( \varepsilon T \) from discretization
\[
\geq \frac{1}{9} \sum_{t} \text{rev}(\mathbf{p^*}, v^+) - \varepsilon T - n \sqrt{\frac{\log \frac{1}{\delta}}{T}}
\]
\[
\sum \frac{1}{2} \sum \text{rev}(\tilde{x}_t, v^+) - \varepsilon T - n \sqrt{\frac{T \log \varepsilon}{T}}
\]

\[\varepsilon = \frac{1}{\sqrt{T}} \rightarrow \text{Regret} \leq O\left( n \sqrt{T \log T} \right)\]