CS 6840 Algorithmic Game Theory September 25, 2024

Lecture 13: Auctions in Bayesian Setting

Instructor: Eva Tardos *Scribe: Shayan Ranjbarzadeh*

1 Recap

We studied two auctions last time, first price auction and all pay auction. Today we want to add the uncertainty of opponents to them.

For special strategy $s_i^* = \frac{1}{2}v_i$ and arbitary strategy s we have seen:

$$\sum_i u_i(s_i^*, s_{-i}) \geq \frac{1}{2} \text{ OPT SW} - Rev(s) \tag{1}$$

If σ is mixed Nash equilibrium:

$$\text{for every player i: } \mathbb{E}_{s\sim\sigma}(u_i(s)) \geq \mathbb{E}_{s\sim\sigma} u_i(s_i^*, s_{-i})$$

by combining the above two inequalities (taking expectation from the first one) we have:

$$\sum_i \mathbb{E}_{s\sim\sigma}(u_i(s)) \geq \sum_i \mathbb{E}_{s\sim\sigma}(u_i(s_i^*, s_{-i})) \geq \frac{1}{2} \text{ OPT SW} - \mathbb{E}_{s\sim\sigma}(Rev(s))$$
$$\implies \mathbb{E}_{s\sim\sigma}(SW(s)) = \mathbb{E}_{s\sim\sigma}(u_i(s)) + \mathbb{E}_{s\sim\sigma}(Rev(s)) \geq \frac{1}{2} \text{ OPT SW}$$

2 Bayes Version of First Price Auction

Let P be the set of all participants who might show up to the auction and $\mathcal{F}$ be a known distribution over 2^P. Suppose $v \sim \mathcal{F}$ is the value vector of a fixed subset of participants who show up to the auction. Then σ is **Bayes Nash equilibrium** iff for all player i and pure strategies s':

$$\mathbb{E}_{v\sim\mathcal{F}}\mathbb{E}_{s\sim\sigma}(u_i(v, s)) \geq \mathbb{E}_{v\sim\mathcal{F}}\mathbb{E}_{s\sim\sigma}(u_i(v, s_i', s_{-i}))$$

where $u_i(v, s)$ is the utility of player i when playing strategy s and the set of participants is v.

The last session proof is still true for this version:

$$\sum_i \mathbb{E}_{v\sim\mathcal{F}}\mathbb{E}_{s\sim\sigma}(u_i(v, s)) \geq \sum_i \mathbb{E}_{v\sim\mathcal{F}}\mathbb{E}_{s\sim\sigma}(u_i(v, s_i', s_{-i}))$$
$$\geq \frac{1}{2}\mathbb{E}_{v\sim\mathcal{F}}\mathbb{E}_{s\sim\sigma}[\text{OPT } SW(v)] - \mathbb{E}_{v\sim\mathcal{F}}\mathbb{E}_{s\sim\sigma}[Rev(v, s)]$$

Note that this is implied by (1) with taking expectation.
Similarly, for no-regret learning and using the following inequality:

$$\sum_t \mathbb{E}_{v^t\sim\mathcal{F}}\mathbb{E}_{s^t\sim\sigma} u_i(v^t, s^t) \geq \sum_t \mathbb{E}_{v^t\sim\mathcal{F}}\mathbb{E}_{s^t\sim\sigma} u_i(v_i^*/2, s_{-i}^t) - \text{Reg}$$

we can derive the below worst-case guarantee using (1):

$$\sum_t \mathbb{E}_{v^t\sim\mathcal{F}}\mathbb{E}_{s^t\sim\sigma} SW(s^t) \geq \sum_t \mathbb{E}_{v^t\sim\mathcal{F}}\mathbb{E}_{s^t\sim\sigma} \frac{1}{2}[\text{OPT } SW(v^t)] - n\cdot \text{Reg}$$

So the same result extends with a small loss for regret.

3 All Pay Auction

Recall that we defined a special strategy s^* for all pay as:

$$\begin{cases} s_i^* \sim \text{unif}[0, v_i] & i = \overbrace{\arg\max_j v_j}^{k} \\ s_i^* = 0 & \text{otherwise} \end{cases}$$

This way for an arbitrary strategy s and $i \neq k$ we have $u_i(s_i^*, s_{-i}) = 0$, and for k we showed that

$$\mathbb{E}[u_k(s_k^*, s_{-k})] \geq v_k/2 - Rev(s)$$

Moreover, in no-regret learning, for every player i we have:

$$\sum_t u_i(s^t) \geq \sum_t u_i(x, s_{-i}^t) - \text{Reg} \quad \text{for all } x \in [0, v_i]$$

By taking expectation over the random alternate bid for the top player we get:

$$\sum_t u_i(s^t) \geq \sum_t \mathbb{E}_{x\sim[0,v_i]}[u_i(x, s_{-i}^t)] - \text{Reg} \tag{2}$$

Again using our framework:

$$\begin{aligned} \sum_t SW(s^t) &= \sum_t \sum_i u_i(s^t) + \sum_t \text{Rev}(s^t) \\ &\geq \sum_t \sum_i \mathbb{E}[u_i(s_i^*, s_{-i}^t)] - n\cdot\text{Reg} + \sum_t \text{Rev}(s^t) && \text{(By (2))} \\ &\geq \frac{T}{2}\max_j v_j - n\cdot \text{Reg} && \text{(By (1))} \end{aligned}$$

which implies a worst case guarantee for no-regret learning.

3.1 Bayes Version of All Pay

Although we could get the same guarantee as first-price for the classic setting, the same argument doesn't seem to work for Bayes version of all pay. This is because s^* depends on other players' values. What they should regret depends on if they are the $\arg\max_j v_j$. No-regret only guarantees no-regret for fixed strategies. It's OK to make it randomized, but the same randomization all the time only. As the set of participants is different at each iteration, and s^* depends on this set, learning becomes ineffective.