CS 6840 Algorithmic Game Theory May 1, 2020

Lecture 31: Limitation of Learning

Instructor: Eva Tardos *Scribe: Jun-You Liu , Yixuan Gao*

Since previous several lectures are talking about how learning improve one's strategy in auction games, today's topic will focus on the **limitation of learning**. By providing an example with the below settings, we are going to see why learning is difficult.

Example: It is an unit demand bidding games, where player i has v_{ij} for a set A. $v_i = \max_{j \in A} v_{ij}$.

Claim: learning to bid in first or second price auction with m items is NP-hard.

Reminder: If bidding on d items only (e.g. $d = m$) and discretizing into multiple of δ, the number of possible bids, $\binom{m}{d}\delta^{-d} \sim (\frac{m}{\delta})^d$. It is maybe okay when d is small, but clearly it is exponential in d. The NP-hardness tells us that we should expect our algorithm to be exponential in some parameter. When considering the *Follow the Leader* strategy, we could get rid of the exponential factor, (based on last two lectures). However, this will only work when we value all items, instead of valuing one item. (Note: unless P=NP, no polynomial time strategy will work.)

Learning is NP-hard

Let's get back to the setting of the example at the beginning to prove learning is NP-hard. We assume a super special case with following conditions to setup

- For player i, its value is v for every item.
- Opponent's bidding strategy

$$b_{-i} = \begin{cases} 1, \text{for a set of } d \text{ items} \\ v^{'} >> v, \text{otherwise} \end{cases} \tag{1}$$

- There are k possible sets, $T_1, T_2, ..., T_k$, that can be the set of d items that the opponent will bid 1 on. Each of them can be chosen uniformly random, which is probability $\frac{1}{k}$ each.

Question: If the opponent is guaranteed to bid this way, can we learn to bid against this opponent?

This questions is more like a computational problem instead of a learning problem since we have already been told what the opponent will do, so the input of this problem is the list of sets $T_1, ..., T_k$ and the values $1 < v < v^{'}$. Besides, we will ignore the learning error for now, we only want to find the truly best bid given the fact provided in the assumptions.

Answer: Let's define the set S to be the items we would like to bid on.

- In second price:

$$b_j = \begin{cases} v, j \in S \\ 0, \text{otherwise} \end{cases} \tag{2}$$

  If I don't want an item at all (item $j \notin S$), I will bid 0. If we want that item (item $j \in S$), we will bid $b_j = v$. In this case, if the opponent bid $v^{'}$, we lose, but it is fine. If the opponent bid 1, we win and only need to pay 1

- In first price:

$$b_j = \begin{cases} 1+\epsilon, j \in S \\ 0, \text{otherwise} \end{cases} \tag{3}$$

we want to bid against the opponent on items we want (items $j \in S$) with $1+\epsilon$ is enough to win and also pay less.

Utility of bidding above 1 on set S is:

$$\mathbb{E}(u(S)) = \frac{1}{k}\sum_i [v(\text{if } T_i \cap S \neq \emptyset) - |T_i \cap S|] \tag{4}$$

For simplicity, we assume that

- $|T_i| = d, \forall i$ (assumed above)
- $v = 2kd$

Then, we can have an example strategy: $S = \bigcup_i T_i \Rightarrow u(S) = v - \frac{1}{k}\sum_i |T_i \cap S| = v - d.$

Claim: optimal S has $T_i \cap S \neq \emptyset$ for all i.

Proof of Claim: Suppose $S \cap T_i = \emptyset$ for one $i \Rightarrow u(S) \leq \frac{1}{k}\sum_i v(\text{if } T_i \cap S \neq \emptyset) \leq \frac{k-1}{k}v = v - 2d$

Remaining Optimization problem: Given $T_1...T_k$, find S such that $S \cap T_i \neq \emptyset$ for all i, and minimize $\sum_i |S \cap T_i|$

The above problem is a hitting set problem with a different objective function.

Hitting set: Given $T_1....T_k$ find S such that $S \cap T_i \neq \emptyset$ for all i and $\min |S|$.

As we know, this is a NP-complete problem. However, it is a little bit different from what we want to use, so we make another special assumption to construct this into regular hitting set, which is a special case of hitting set.

Regular hitting set: $|T_i| = d$ for all i, and $\#\{i | j \in T_i\} = r$ for all elements j.

Claim: Regular hitting set is NP-complete, and $\frac{1}{2}\log r$ approximation is also NP-hard.

With all claims we have, we get that given $T_1, ...T_k$, $\sum_i |T_i \cap S| = \sum_{j \in S} \#\{i | j \in T_i\} = r|S|$ (by regularity).

Therefore, under regularity, our problem is equivalent to hitting set, and so we just proved that finding optimal bidding strategy is at least as hard as finding solution to regular hitting set.

Concluding issues

1. Learning does not find the answer "exactly" but "approximately". Learning finds a solution $\geq (1-\epsilon)Opt - v\frac{\log x}{\epsilon}$. With $\epsilon = \sqrt{\frac{\log x}{T}}$ this gives us a value $\geq Opt - 2v\sqrt{T \log x}$, which is a small enough error that getting this close is not possible without approximating the regular hitting set well.

2. Note what we really prove is also that it is hard to run the "follow the leader" algorithm. Given history $T_1...T_k$, finding the best strategy (bid) given this history is exactly what we proved to be NP-Hard.