CS 6840 Algorithmic Game Theory April 26, 2020

Lecture 32: Learning via Follow the Perturbed Leader cont.

Instructor: Eva Tardos *Scribe: Megan Le, Quintessa Qiao*

Topics

- Today: Follow the Perturbed Leader
- Wednesday: Other ways to allocate items
- Limits of learning
- Playing against the learner

Follow the Perturbed Leader

Summary

Assume there is a learner facing an unknown utility. The learner has k strategies where $k \in S$. Utility is defined as $u^t(s) \in [0,1]$ for time t where $t = 1, ..., T$.

Algorithm: Select strategy s^t such that

$$s^t = argmax \sum_{\tau=1}^{t-1} u^\tau(s) + \sigma_s$$

Σ_s is defined as the random # throws of dice until H shows up where H has prob σ independent for all s.

We proved for all sequences of $u^t(s)$ values (fixed) that

$$E_\sigma(\sum_{t=1}^{T} u^t(s^t)) \geq (1-\epsilon) max_s \sum_{t=1}^{T} u^t(s) + E_\sigma(max_s \sigma_s)$$

However, there is some trouble in using this. If the opponent can react to the choice of σ, $u^t(s)$ will depend on what σ is. After a while, the opponent will figure out what σ is. If σ is not changed, eventually what the learner is playing is deterministic.

Improved version: Re-randomize σ with each step which will fix the deterministic problem.

Claim: Above bound extends with this change to games where $u^t(s)$ can depend on previous σs

Proof: We can use linearity of expectations. We will be indexing σ with time now which gives us the new rule

$$s^t = argmax_s \sum_{\tau=1}^{t-1} u^\tau(s) + \sigma_s^t$$

Start with our summation of expectation

$$\sum_{t=1}^{T} E_\sigma(u^t(s^t))$$

Because each iteration uses same distribution, let's call it σ. This allows us to rewrite the above as

$$\sum_{t=1}^{T} E_\sigma(u^t(s^t)) = E_\sigma(\sum_{t=1}^{T} u^t(s^t))$$

And then using the above

$$\sum_{t=1}^{T} E_\sigma(u^t(s^t)) = E_\sigma(\sum_{t=1}^{T} u^t(s^t)) \leq (1-\epsilon)E_\sigma(max_s \sum_{t=1}^{T} u^t(s) - E_\sigma(max\sigma_s))$$

Note that in

$$(1-\epsilon)E_\sigma(max_s \sum_{t=1}^{T} u^t(s) - E_\sigma(max\sigma_s))$$

u^t may change reacting to $\sigma^1...\sigma^{t-1}$

Fact: (see Friday's notes) $E(max_s\sigma_s) \leq 2\epsilon^{-1}logk$

Summary: In a Game

Player i uses Follow the Perturbed Leader learning where i is the only player learning.

$$E(\sum_{t=1}^{T} u_i(s^t)) \geq (1-\epsilon)E(max_{s_i} \sum_{t=1}^{T} u_i(s_i, s^t_{-i})) - 2\frac{logk_i}{\epsilon})$$

where $s^t = (s^t_1...s^{\#}_n)$ and $k_i = \#$ of strategies of player i

In what way is this better than multiplicative weight?

- Bound is the same.

Question: Is there a way to run the algorithm more efficiently?

Application 1

Let's look at an auction with unit demand. In unit demand, each participant only values one item or $v_i(A) = max_{j\in A}v_{ij}$. Additionally, participants may only bid on one item. Assume discrete bids are used where bids are multiple of δ and $v_{ij} \in [0,1]$. The number of bids is $k = m * \delta^{-1}$ where m is the number of items.

Algorithm: Choose $m\delta^{-1}$ random σ_s each step
Note: There is no improvement over multiplicative weight in this application.

Application 2

Let's now look at an auction with d-demand. In d-demand, each participant may value up to d items or $v_i(A) = max_{A' \subseteq A_{|A'| \leq d}} \sum_{j \in A'} v_{ij}$. Participants may only bid on d items. Bid multiples of δ are used. Additionally, the number of bids can be defined as $\binom{m}{d}\delta^{-d} = k_i$.

Regret Guarantee:

- $E(\sum u^t(s^t) \geq (1-\epsilon)E(\max_{s_i} \sum u^t(s_i, s^t_{-i})) - 2\frac{\log k_i}{\epsilon} = (1-\epsilon)E(\max_{s_i} \sum u^t(s_i, s^t_{-i})) - 2\frac{d\log(m\delta^{-1})}{\epsilon}$
- # random σ_s needed: $\binom{m}{d}\delta^{-d} \sim m^d\delta^{-d}$

Proposed alternate: Pick σ for all bids, on each bid separately. There's only $m\delta^{-1}$ of them.

Strategy s = select set of items A and bid on them where $\sigma_s = \sum_{i \in A} \sigma_{s_i}$
Add random choices over d items in bids

Claim: Same regret bound works and avoids exponential dependence of d in running algorithm

Proof of Claim added in notes

Recall we assumed that the utilities are $u(s) \in [0,1]$. To make notation simpler it seems better to assume each value $v_{ij} \in [0,1]$, so then values $v_i(A) \in [0,d]$. With this change in normalization, the follow the perturbed leader above would add σ_s that is d times the number of coin flips till we get a H, and get a guarantee that is

$$E(\sum u_i^t(s^t) \geq (1-\epsilon)E(\max \sum u_i^t(s_i, s^t_{-i})) - O(d\frac{\log k_i}{\epsilon}) = (1-\epsilon)E(\max_{s_i} \sum u_i^t(s_i, s^t_{-i})) - O(d^2\frac{\log(m\delta^{-1})}{\epsilon})$$

We will show that the same guarantee can be achieved by each possible bid on an item j separately and independently choosing a σ_{j,b_j} that is the number of coin flips till a head comes, but using a different parameter ϵ', which we will choose later, and then defining σ_s for a strategy b that has bids for a subset A of d items as

$$\sigma_s = \sum_{j \in A} \sigma_{j,b_j}$$

To prove the claimed regret guarantee, we need to extend the two claims from last lecture to this case. Note that both claims use an ϵ that will not be the same as the parameter ϵ' using in defining the σ's and they both have an extra d multiplier in the bound, as everything is scaled up by a factor of d with the utilities in the $[0,d]$ range.

1. $E(\max \sigma_s) \leq O(d\log(m\delta^{-1}))$
2. In each iteration t, with probability at least $(1-\epsilon)$ the choice of the "be the leader" algorithm that has access to the iteration t utilities has at least a d difference between the best and the next best strategy (this way the "be the leader" and the actual follow the leader make the same choice).

We will prove bounds with ϵ' and set ϵ later.

1. Recall from last time that with some K random σ_s values $E(\max \sigma_s) \leq O(\epsilon'^{-1} \log K)$. We used $m\delta^{-1}$ random σ's and for each strategy we are adding d of them, so we get

$$E(\max \sigma_s) \leq d \cdot E(\max \sigma_{j,b_j}) = O(d\epsilon'^{-1} \log(m\delta^{-1}))$$

2. to prove the second point, we will extend the proof from the previous class. Consider an outcome of coin flips in the follow the leader algorithm that a bid b is the strategy chosen: outcomes of all coin flips determining the σ_{j,b_j} for all bids not in the chosen one and the initial tails that lead guarantee that the bid-vector chosen is the one with maximum value. After this set of coin flip outcomes, it is guaranteed that b will be the chosen bid in the "be the perturbed leader" algorithm. If all bids b_j in the chosen bid vector add one extra T before any H than the difference between this bid b and every other bid at at least d, and so the follow the perturbed leader will also choose this bid. This event has $(1-\epsilon')^d$ probability. Using the same notation as last class with b^t the bid that would have been chosen with the utility at time t included, and $\bar{b}^t$ the bid chosen by the true perturbed follow the leader, we get

$$\begin{aligned} E(\sum_t u^t(\bar{b}^t)) &\geq E(\sum_{t:\bar{b}^t=b^t} u^t(b^t)) = (1-\epsilon')^d E(\sum_t u^t(b^t)) \geq (1-\epsilon')^d \max_b E(\sum_t u^t(b)) - E(\max(\sigma_b)) \\ &= (1-\epsilon')^d \max_b E(\sum_t u^t(b)) - O(d\epsilon'^{-1} \log(m\delta^{-1})) \end{aligned}$$

Now set $\epsilon' = \epsilon/d$. With this choice we get $(1-\epsilon')^d \geq (1-d\epsilon') = (1-\epsilon)$. Using this we get

$$E(\sum_t u^t(\bar{b}^t)) \geq (1-\epsilon) \max_b E(\sum_t u^t(b)) - O(d^2\epsilon^{-1} \log(m\delta^{-1}))$$

As claimed.