CS 6840 Algorithmic Game Theory April 22, 2020

Lecture 30: Follow the Leader (or be the leader) as Learning

Instructor: Eva Tardos *Scribe: Shir Maimon, Renee Mirka*

Introduction

In previous lectures we have discussed when all players use a no-regret learning strategy. We have shown that this leads to a high quality of outcome (in terms of social welfare). Today we begin talking about what learning can do before moving into the limits of what learning can do (unless P=NP) in future lectures.

Today's Setup

We begin by considering "Follow the Leader" learning discussed by Robinson in '56. The idea is to "do what would have been best in the past".

Setup:

- each learner has k options of $s \in S$
- at time t, $u^t(s)$ is the utility of the learner for using s
- for player i, $u_i^t(s) = u_i(s, s_{-i}^t)$ where s_{-i}^t is what the others did
- we assume $0 \leq u^t(s) \leq 1$

Our hope is for an algorithm choosing s^t at time t such that

$$\sum_{t=1}^{T} u^t(s^t) \geq (1-\epsilon) \max_s \sum_{t=1}^{T} u^t(s) - O(\frac{\ln k}{\epsilon}).$$

The sum on the left is the algorithm's utility and ϵ is a sensitivity parameter. We can also consider this inequality in expectation to allow for randomized choices.

If our algorithm chooses s^t such that

$$\sum_{t=1}^{T} u^t(s^t) \geq \max_s \sum_{t=1}^{T} u^t(s)$$

then we say that we have 0 regret.

Follow the Leader

In follow the leader learning, we let $s^t = \arg\max_s \sum_{\tau=1}^{t-1} u^\tau(s)$.

Bad Example (for any deterministic algorithm):

Let the utilities for s_1 be 1,0,1,0,1,0,... and for s_2 be .6,1/2,1/2,1/2,1/2,1/2... In the follow the leader strategy the learner will alternate between s_1 and s_2. However, the learner will be choosing s_1 when it's utility is 0, and s_2 when s_1's utility is 1. Therefore, at time T, the total utility is about $T/4$, whereas fixing either strategy would give total utility of about $T/2$ at time T. This example illustrates the need to randomize.

Follow the Perturbed Leader

Follow the perturbed leader learning is a slight modification to follow the leader. In this scenario, select a random ξ_s for all s independently. Then at time t, choose the strategy such that $s^t = \arg\max_s(\sum_{\tau=1}^{t-1} u^\tau(s) + \xi_s)$.

To show that this randomization is effective we will consider the following "imaginary algorithm": let $s^t = \arg\max_s \sum_{\tau+1}^{t} u^t(s)$. Note that this algorithm is not necessarily possible as it requires knowledge of $u^t(s)$ before bidding. Our goal is to show that the imaginary algorithm with noise ξ added works well, and then that the true algorithm with ξ works approximately as well as the imaginary algorithm with ξ.

Regret of imaginary algorithm

First, we show that the imaginary algorithm has 0 regret. We can do this by induction on t.

If $t = 1$ then

$$s^t = \arg\max_s \sum_{\tau=1}^{t} u^\tau(s) = \arg\max_s u^1(s)$$

so

$$u^1(s^1) = \max_s u^1(s).$$

Suppose there is 0 regret at time t. Then we have:

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

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

$$\geq \sum_{\tau=1}^{t} u^\tau(s^{t+1}) + u^{t+1}(s^{t+1})$$

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

$$= \max_s \sum_{\tau=1}^{t+1} u^\tau(s)$$

where the inequality is by the induction hypothesis, and the last equality is by our choice of s^{t+1}. Thus the algorithm has 0 regret.

No we want to find what happens when we add noise ξ_s. That is, let

$$s^t = \arg\max_s(\sum_{\tau+1}^{t} u^t(s) + \xi_s).$$

Then we can use a similar proof to show that there is an error of at most $\max_s(\xi_s)$, assuming every $\xi_s \geq 0$. We will show by induction that

$$\sum_{\tau=1}^{t} u^\tau(s^\tau) + \max \xi_s \geq \max_s(\sum_{\tau=1}^{t} u^\tau(s) + \xi_s).$$

For the base case, if $t = 1$, then

$$s^t = \arg\max_s(\sum_{\tau=1}^{t} u^\tau(s) + \xi_s) = \arg\max_s(u^1(s) + \xi_s)$$

so

$$u^1(s^1) + \xi_1 = \max_s(u^1(s) + \xi_s)$$

and

$$u^1(s^1) + \max_s \xi_s \geq \max_s(u^1(s) + \xi_s).$$

Now suppose that

$$\sum_{\tau=1}^{t} u^\tau(s^\tau) + \max \xi_s \geq \max_s(\sum_{\tau=1}^{t} u^\tau(s) + \xi_s).$$

Then

$$\begin{aligned}
&\sum_{\tau=1}^{t+1} u^\tau(s^\tau) + \max \xi_s \\
&= \sum_{\tau=1}^{t} u^\tau(s^\tau) + \max \xi_s + u^{t+1}(s^{t+1}) \\
&\geq \max_s(\sum_{\tau=1}^{t} u^\tau(s) + \xi_s) + u^{t+1}(s^{t+1}) \\
&\geq \sum_{\tau=1}^{t} u^\tau(s^{t+1}) + \xi_{s^{t+1}} + u^{t+1}(s^{t+1}) \\
&= \sum_{\tau=1}^{t+1} u^\tau(s^{t+1}) + \xi_{s^{t+1}} \\
&= \max_s(\sum_{\tau=1}^{t+1} u^\tau(s) + \xi_s).
\end{aligned}$$

So for all t,

$$\sum_{\tau=1}^{t} u^\tau(s^\tau) + \max \xi_s \geq \max_s(\sum_{\tau=1}^{t} u^\tau(s) + \xi_s)$$

$$\sum_{\tau=1}^{t} u^\tau(s^\tau) \geq \max_s(\sum_{\tau=1}^{t} u^\tau(s) + \xi_s) - \max \xi_s$$

so if we let $\bar{s} = \arg\max_s(\sum_{\tau=1}^{t} u^\tau(s))$

$$\sum_{\tau=1}^{t} u^\tau(s^\tau) \geq \sum_{\tau=1}^{t} u^\tau(\bar{s}) + \xi_{\bar{s}} - \max \xi_s \geq \sum_{\tau=1}^{t} u^\tau(\bar{s}) - \max \xi_s = \max_s(\sum_{\tau=1}^{t} u^\tau(s)) - \max \xi_s.$$

Next class we will show how to use this to analyze the read algorithm.