CS295 Introduction to Algorithmic Game Theory

Instructor: Ioannis Panageas Scribed by: Yu-Chen Wang

Lecture 4. Online Learning and a proof of Minimax Theorem

1 Introduction

In this lecture, we will discuss about connections between online learning and Minimax Theorem [2]. We will first introduce Experts Game and have an algorithm called Weighted Majority Algorithm to solve it. We will then have a generalization version of Expert Game and introduce another algorithm called Multiplicative Weights Update Algorithm [3] to solve it. We will also introduce No-Regret Learning. At last, we will prove the Minimax Theorem with No-Regret Learning.

2 Expert Game and Weighted Majority Algorithm

Firstly we take a look at the following game.

Definition 2.1 ***Expert Game*** *We have a game of T rounds, for each round $t = 1...T$, we have to choose between two alternatives A, B. The rule of game is the following:*

- *Choose A or B according to some rules.*
- *One of the alternatives realizes.*
- *If we make a correct choice, we will not be penalized, otherwise we will lose 1 point.*
- *There are n experts there to give us some hints. In each day, they will give us some recommendations.*

Our goal in this game is to make as less mistakes as possible. Suppose we know there is a best expert, then it is easy to know that our best strategy is to follow the advice of this expert since the expectation of penalty will be minimized for each of the round. However, in this game we do not know who is the best expert at the first place. We need to introduce online learning to learn who is the best expert. Because of this we introduce the **Weighted Majority Algorithm** [1] as in 2.

Now we have our first algorithm, next we will show that the number of mistakes we will make if we applied this algorithm is bounded.

Theorem 2.1 *Let M_T, M_T^B be the total number of mistakes the algorithm shown in 2 and the best expert make until step T, respectively. It holds that:*

$$M_T \leq 2(1+\epsilon)M_T^B + \frac{\log n}{\epsilon} \tag{1}$$

Algorithm 1 Weighted Majority Algorithm

Require: $n > 0$, $T > 0$, $0 < w_i \le 1$, $0 < \epsilon < 1$

Ensure: $w_i = 1$

for $t = 1...T$ **do**

  if $\sum_{i \text{ choose } A} w_i^{t-1} \ge \sum_{i \text{ choose } B} w_i^{t-1}$ **then**

    We choose A, otherwise we choose B.

  end if

  For all experts that make a mistake, we have $w^t = (1-\epsilon)w^{t-1}$, otherwise we have $w^t = w^{t-1}$

end for

Proof:

Let's define the following function $\phi_t = \sum_i w_i^t$. According to the initialization and update rules of 2, we will have that $\phi_0 = n$ and $\phi_t \le \phi_{t-1}$.

We will have the following lemma.

Lemma 2.2 *In the iteration t of Algorithm 2, if we make a mistake, we will have:*

$$\phi_t \le \left(1 - \frac{\epsilon}{2}\right)\phi_{t-1} \tag{2}$$

According to the update rule of 2, we will have:

$$\phi_t = (1-\epsilon)\sum_{\text{Experts mistake}} w^{t-1} + \sum_{\text{Experts no mistake}} w^{t-1} \tag{3}$$

Since we make a mistake, we know $\sum_{\text{Experts mistake}} w^{t-1} \ge \phi_{t-1}/2$, thus we have:

$$\phi_t \le \left(1 - \frac{\epsilon}{2}\right)\phi_{t-1}$$

Since we make M_T mistakes until iteration T, apply 2.2 M_T times and we will have:

$$\phi_T \le \left(1 - \frac{\epsilon}{2}\right)^{M_T}\phi_0 = \left(1 - \frac{\epsilon}{2}\right)^{M_T} N \tag{4}$$

Since M_T^B is the number of mistakes that the best expert makes. We will know that in iteration T, the weight of the best expert is $(1-\epsilon)^{M_T^B}$. Since ϕ_T is the sum of weights, we will have:

$$\phi_T > (1-\epsilon)^{M_T^B} \tag{5}$$

Then we will have:

$$(1-\epsilon)^{M_T^B} < \left(1 - \frac{\epsilon}{2}\right)^{M_T} N \tag{6}$$

Taking the log, we will have:

$$\log{(1-\epsilon)}M_T^B < M_T \log{(1-\epsilon/2)} + \log n \tag{7}$$

Since we have the following inequality $-x - x^2 < \log{(1-x)}$ and $\log{(1-x/2)} > -x/2$, we will have:

$$(-\epsilon - \epsilon^2)M_T^B < -\frac{\epsilon}{2}M_T + \log n \tag{8}$$

We clean it up and we will finish the proof.

∎

3 A Generalization of Expert Game and Multiplicative Weights Update

Now suppose we have the following game setting.

Definition 3.1 *Suppose we have T rounds in this game. At each time step $t = 1...T$, we have:*

- *Player choose $x_t \in \Delta_n$, where Δ_n is the space of strategies.*
- *Adversary choose $u_t \in [-1, 1]^n$.*
- *Player gets payoff $x_t^T u_t$ and observe u_t.*

The goal of player is to minimize the following object function:

$$\frac{1}{T}\left[\max_{x\in\Delta_n}\sum_{t=1}^{T} x^T u_t - \sum_{t=1} x_t^T u_t\right]$$

*Note that we also call this object function **Regret**.*

Definition 3.2 ***No Regret Algorithm*** *If we have Regret $\to 0$ as $T \to \infty$, the algorithm is called **no-regret**.*

Firstly we want to show the following lemma:

Lemma 3.1 *The object function of the game described in 3.1 have:*

$$\frac{1}{T}\left[\max_{x\in\Delta_n}\sum_{t=1}^{T} x^T u_t - \sum_{t=1} x_t^T u_t\right] = \frac{1}{T}\left[\max_{i^*\in[n]}\sum_{t=1}^{T} u_{t,i} - \sum_{t=1} x_t^T u_t\right]$$

Proof Outline:
Suppose i^* is the index of best expert. By definition, we will have:

$$\sum_{t=1}^{T} u_{t,i^*} \geq \sum_{t=1}^{T} u_{t,i}$$

Since x is a probability expectation, it is obvious that always choosing the best expert is the best we can do if we use a same strategy for all rounds. In this way, this object function becomes can we have a series of strategies that make our pay off close to the performance of the best expert.

■

Now consider the following algorithm.

Similar to Weighted Majority Algorithm, we will show that our payoff given by Weighted Majority Algorithm described in 3 is bounded. That is, we will show the following theorem.

Theorem 3.2 *The algorithm shown in 3 holds that:*

$$\frac{1}{T}\sum_{t} u_t^T p^t \geq \max_{x}\sum_{t} x^T u_t - \frac{\log n}{\epsilon T} - \epsilon$$

Algorithm 2 Weighted Majority Algorithm

Require: $n > 0$, $T > 0$, $0 < w_i \leq 1$, $0 < \epsilon < 1$

Ensure: $w_i = 1$

for $t = 1...T$ **do**

Choose an action with respect to a pdf $\mathcal{D} = \{p_i^t = w_i^t / \sum_i w_i^t\}$

for each i that gives payoff $u_{t,i}$ **do**

$w_i^{t+1} = w_i^t(1 + \epsilon u_{t,i})$ ▷ Update the weights

end for

end for

Proof:

Let's define the following function $\phi_t = \sum_i w_i^t$.

For ϕ_{t+1}, we have that:

$$\phi_{t+1} = \sum_i w_i^{t+1} = \sum_i w_i^t(1 + \epsilon u_{t,i})$$

Since we have $p_i^t = w_i^t / \phi_t$, we will have:

$$\phi_{t+1} = \phi_t \sum_i p_i^t(1 + \epsilon u_{t,i}) \tag{9}$$

Since we have the following inequality: $1 + x \leq e^x$, we have:

$$\phi_{t+1} \leq \phi_t e^{\epsilon u_t^T p^t}$$

Apply this recursive equation repeatedly, we will have:

$$\phi_{t+1} \leq \phi_0 e^{\epsilon \sum_t u_t^T p^t} = n e^{\epsilon \sum_t u_t^T p^t} \tag{10}$$

On the other hand, suppose the index of best expert is i^*, we will have:

$$\phi_T > w_{i^*}^T = \prod_{t=0}^{T}(1 + \epsilon u_{t,i}) \tag{11}$$

We apply the fact that $1 + x \leq e^{x - x^2}$ when x is small. Then we will have that:

$$\phi_T > e^{\epsilon \sum_{t=0}^T u_{s,i^*} - \epsilon^2 \sum_{t=1}^T u_{s,i^*}^2} \tag{12}$$

Combine 10 and 12, we will have that:

$$e^{\epsilon \sum_{t=0}^T u_{s,i^*} - \epsilon^2 \sum_{t=1}^T u_{s,i^*}^2} < n e^{\epsilon \sum_t u_t^T p^t} \tag{13}$$

Taking log from both sides, we will have that:

$$\epsilon \sum_{t=0}^{T} u_{s,i^*} - \epsilon^2 \sum_{t=1}^{T} u_{s,i^*}^2 < \log n + \epsilon \sum_t u_t^T p^t$$

Divide both sides by ϵT and clean it up, we will conclude this theorem.

■

4 A Proof of Minimax Theorem with No-Regret Learning

In the previous sections, we talked about a generalization of experts game and an online learning algorithm to get good results in these games. In this section, we will revisit Von Neumann's Minimax Theorem again and show a proof of it based on No-Regret Learning.

First let's take a look at Minimax Theorem.

Theorem 4.1 ***Minimax Theorem*** *Let A a matrix $n \cdot m$ to be a payoff matrix of a two player game. Δ_n and Δ_m is the set of actions of the row player and the column player, respectively. We have:*

$$\min_{x \in \Delta_n} \max_{y \in \Delta_m} x^T Ay = \max_{y \in \Delta_m} \min_{x \in \Delta_n} x^T Ay$$

Proof:

First we will prove the following inequality:

$$\min_{x \in \Delta_n} \max_{y \in \Delta_m} x^T Ay \geq \max_{y \in \Delta_m} \min_{x \in \Delta_n} x^T Ay \tag{14}$$

Suppose we have $g(y) = \inf_x x^T Ay$, then we know that for all y, we have $x^T Ay \geq g(y)$. Then we have:

$$\sup_y g(y) \leq \sup_y x^T Ay$$

Taking minimum for the right side, we will conclude this inequality.

Next, we want to prove the following inequality:

$$\min_{x \in \Delta_n} \max_{y \in \Delta_m} x^T Ay \leq \max_{y \in \Delta_m} \min_{x \in \Delta_n} x^T Ay \tag{15}$$

Suppose we have the following game setting:

- In each iteration t, player choose the best strategy x^t and y^t.
- Players will play at the same time, that is, they do not know each other's strategy in each iteration, however, they will observe opponents' in the current iteration.

Since A is the payoff matrix of Player 1, the goal of Player A is to find $\max_{x^* \in \Delta_x} x^* Ay$. For the player B, the goal is to find $\max_{y^* \in \Delta_y} xAy^*$. Take a look at $v = \frac{1}{T} \sum_{t=1}^{T} x^t Ay^t$.

Suppose x^t and y^t are from MWU, by the properties of regret, in the place of Player A, we will have:

$$v \geq \frac{1}{T} \max_{x \in \Delta_x} \sum_{i=1}^{t} xAy_t - \epsilon \geq \min_{y \in \Delta_m} \max_{x \in \Delta_n} xAy - \eta$$

In the place of Player B, we will have:

$$v \leq \frac{1}{T} \min_{y \in \Delta_m} \sum_{i=1}^{t} x_t Ay + \epsilon \leq \max_{x \in \Delta_n} \min_{y \in \Delta_m} xAy + \eta$$

Put the above 2 inequalities together, we will have:

$$\min_{y \in \Delta_m} \max_{x \in \Delta_n} xAy - \eta \leq \max_{x \in \Delta_n} \min_{y \in \Delta_m} xAy + \eta \tag{16}$$

If we have $\epsilon \to 0$ and combine equation 14 and 16, we will conclude this theorem.

■

References

[1] N. Littlestone, M.K. Warmuth, *The Weighted Majority Algorithm.* Information and Computation, Volume 108, Issue 2, 1994, Pages 212-261

[2] v. Neumann, J. Zur *Theorie der Gesellschaftsspiele. Math.* Ann. 100, 295–320 (1928).

[3] Arora, Sanjeev, Elad Hazan, and Satyen Kale. *The multiplicative weights update method: a meta-algorithm and applications.* Theory of computing 8, no. 1 (2012): 121-164.