CS 6840 Algorithmic Game Theory Feburary 26, 2020

Lecture 14: No-Regret for k-Players

Instructor: Eva Tardos *Scribe: Victoria Mao and Benjamin Chan*

1 Summarizing the previous 2 lectures

Recall in no-regret algorithm (for a one-player game):
- Strategies $1, ..., n$
- Cost $c_i^t \in [0, 1]$ for each strategy i and time step t

Select i_t at time t, then the cost incurred is:

$$\sum_{t=1}^{T} c_{i_t}^t \leq \min_i \sum_{t=1}^{T} c_i^t + 2\sqrt{T \log n}$$

Since this is a cumulative cost, this shows that the regret per time step is at most $2\sqrt{\log n/T}$.

What about in a multiple-player game, where player j uses no-regret strategy?

- Strategies $1, ..., n$
- Cost that player j pays for his choice of strategy i at time t is $c_j^t = cost_j(i, \mathbf{s_{-j}^t})$
where $\mathbf{s^t}$ is the strategy vector of all players at time t

Example (Traffic routing): G = (V, E) a network graph. Player j chooses path from s_j to t_j as a strategy. Then the number of strategies n = number of $s_j - t_j$ paths $\leq 2^{|E|}$, which is exponential.

To bound the *regret*, note that $logn \leq |E|$. However, this is still not practical because:

- Need to maintain explicit probabilities p_i for all $i = 1, ..., n$.
- Need "full information" of cost c_i^t for all i and t

But here n is exponentially large.

(For now we assume it's practical and the feasibility will be discussed later in the class)

2 No-regret algorithms for more than 2 players

We now expand the discussion to games with more than 2 players. Let there be k players who all play with no-regret algorithms. The two takeaways are as follows:

- "Smoothness" machinery works pretty well for the no-regret setting.
- We may not achieve the Nash, but we can do pretty well.

The Good News Assume that our game is (λ, μ) smooth. Let there be k players, each of whom run a no-regret algorithm. Denote a strategy vector $\mathbf{s} = (s_1, \ldots, s_k)$. Also denote the social-cost minimizing strategy vector $\mathbf{s}^* = (s_1^*, \ldots, s_k^*)$. Denote $c_i(\mathbf{s})$ the cost experienced by player i if the strategy vector $\mathbf{s}$ is applied globally. The social cost, as before, is denoted $SC(\mathbf{s}) = \sum_{i=1}^k c_i(\mathbf{s})$.

Moreover, we play many instances of the game over T timeslots $t = 1, \ldots, T$, and notate for each t the strategy vector at time t, $\mathbf{s^t} = (s_1^t, \ldots, s_k^t)$, comprised of k plays by k people at time t.

Recall the smoothness property, namely:

Definition 1 ((λ, μ)-smoothness)**.** A game (as denoted above) is (λ, μ)-smooth if for any $0 \leq \mu < 1$, and for all $\mathbf{s}$, for some λ:

$$\sum_{i=1}^k c_i(s_i^*, \mathbf{s_{-i}}) \leq \lambda SC(\mathbf{s}^*) + \mu SC(\mathbf{s})$$

Also recall the no-regret property:

Definition 2 (No-Regret)**.** A sequence of plays $\mathbf{s^1}, \ldots, \mathbf{s^T}$ is no-regret with respect to the ith player, for costs $0 \leq c_i(\mathbf{s^t}) \leq 1$ if

$$\sum_{t=1}^T c_i(\mathbf{s^t}) \leq \min_{s_i} \sum_{t=1}^T c_i(s_i, \mathbf{s^t_{-i}}) + R$$

for some regret term R.

Now, the main course:

Theorem 1. *If a game (cost-based) is (λ, μ)-smooth, and all players play no-regret algorithms, after a sequence of plays $\mathbf{s^1}, \ldots, \mathbf{s^T}$, then:*

$$\frac{1}{T} \sum_{t=1}^T SC(\mathbf{s^t}) \leq \frac{\lambda}{1-\mu} SC(\mathbf{s}^*) + \frac{k}{1-\mu} \cdot \frac{R}{T}$$

for regret term R.

Proof. We start with the average cost suffered over all the time steps:

$$\begin{aligned}
\frac{1}{T} \sum_{t=1}^T SC(\mathbf{s^t}) &= \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^k c_i(\mathbf{s^t}) \\
&= \frac{1}{T} \sum_{i=1}^k \sum_{t=1}^T c_i(\mathbf{s^t})
\end{aligned}$$

Applying the no-regret property,

$$\begin{aligned}
\frac{1}{T} \sum_{i=1}^k \left(\sum_{t=1}^T c_i(\mathbf{s^t}) \right) &\leq \frac{1}{T} \sum_{i=1}^k \left(\min_{s_i} \sum_{t=1}^T c_i(s_i, \mathbf{s^t_{-i}}) + R \right) \\
&= \frac{1}{T} \left(kR + \sum_{i=1}^k \min_{s_i} \sum_{t=1}^T c_i(s_i, \mathbf{s^t_{-i}}) \right) \\
&\leq \frac{1}{T} \left(kR + \sum_{i=1}^k \sum_{t=1}^T c_i(s_i^*, \mathbf{s^t_{-i}}) \right)
\end{aligned}$$

$$= \frac{1}{T}\left(kR + \sum_{t=1}^{T}\sum_{i=1}^{k} c_i(s_i^*, \mathbf{s_{-i}^t})\right)$$

Now, applying the smoothness property:

$$\leq \frac{1}{T}\left(kR + \sum_{t=1}^{T}\left(\lambda SC(\mathbf{s}^*) + \mu SC(\mathbf{s^t})\right)\right)$$

Rearranging:

$$\frac{1}{T}(1-\mu)\sum_{t=1}^{T} SC(\mathbf{s^t}) \leq \lambda SC(\mathbf{s}^*) + \frac{kR}{T}$$

$$\frac{1}{T}\sum_{t=1}^{T} SC(\mathbf{s^t}) \leq \frac{\lambda}{1-\mu} SC(\mathbf{s}^*) + \frac{kR}{T(1-\mu)}$$

which proves the theorem.

■

Let's briefly synthesize the theorem. The quantity on the left is simply the average social cost of the strategy vectors over time. The first term on the right, $\frac{\lambda}{1-\mu}SC(\mathbf{s}^*)$, says that we're nearly as good as the Nash, and finally we can imagine the term $\frac{R}{T}$ as the averaged regret error. We note as an aside that we can similarly apply these techniques to utility games.

What is this quantity that they are converging to? As we now see, learning in games converges to something that is not the Nash. For now, we will build some intuition here with an example, Rock, Paper, Scissors.

Example (Rock-Paper-Scissors Game):

	Rock	Paper	Scissors
Rock	0	-1	1
Paper	1	0	-1
Scissors	-1	1	0

In this 2-player zero-sum game, the values in the table denote the score that row player wins, and for column player, the negative of each value.

The Nash is that both row player and column player play $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ for (R, P, S) independently.

But in fact, the game may converge to a uniform distribution ($\frac{1}{6}$ each) over the 6 non-diagonal cells in the table (those with value 1 and -1, but not 0). Just imagine if at the beginning, the row player messed up the Nash a little bit and plays more R, then the column player would play more P, again the row player plays more S... Finally, the game converges to $\frac{1}{6}$ each for non-diagonal cells.

We observe that this distribution is also $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ for each player. But different from Nash, here the strategies of the two players are correlated - not independent!
In next lecture, we will talk about correlated Nash.