CS 6840: Algorithmic Game Theory **Spring 2017**

Lecture 14: February 27

Lecturer: Thodoris Lykouris *Scribe: Nitin Shyamkumar*

14.1 Partial Information Feedback Algorithms

14.1.1 Overview

In Lecture 11, we saw that if players use Low Approximate Regret, then we have outcomes that are approximately efficient. In Lecture 12, we saw that there exist algorithms such as multiplicative weights algorithms that give this guarantee. However, these algorithms require players to have full information feedback - that is, access to the information they would have received with different strategies. We will now provide a method to turn full information algorithms into partial information feedback with low approximate regret. We use the following notation.

- $\mathcal{A}_i$:= action set for player i
- N:= number of actions ($N = |\mathcal{A}_i| = |\{a_{i1}, a_{i2}, ..., a_{in}\}|$). These actions are also called arms (from bandit problems in the learning context).
- $p_i^t(a)$:= probability that player i selects action a at round t.

14.1.2 General Protocol for Learning

The player will select a distribution over the arms.

1. Players select distribution over arms $\{a_{ij}\}$

   with $p_i^t(a)$:= probability that player i selects action a at round t.

2. Adversary selects costs for each action: $c_i^t(a)$

   In the context of games, we use $c_i^t(a) = c_i^t(a, a_{-i}^t)$

3. Player i draws action $a_i^t \sim p_i^t$. Note that costs are selected before player observes action.

4. Player i incurs cost $c_i^t(a_i^t)$.

5. Player i observes feedback.

   In full information context, this means player i observes $c_i^t(a) \forall a \in \mathcal{A}_i$.

   In partial information context, player i observes only $c_i^t(a_i^t)$, or "bandit feedback."

We now drop the subscript i as we will be implicitly discussing the player i.

14.1.3 Full Information to Partial Information Context

Recall the full information low approximate regret bound

$$\forall a^* : \sum_t \sum_a p^t(a) c^t(a) \leq (1+\epsilon) \sum_t c^t(a^*) + \frac{\ln N}{\epsilon}$$

That is, summing over all rounds, the expected cost at every step is within a multiplicative factor of $1+\epsilon$ of the cost incurred by selecting the optimal action plus an additional additive term.

Note that we derived these bounds assuming $\forall a, c^t(a) \leq 1$. We could also have $\forall a, c^t(a) \leq L$ and the additive error would be $\frac{\ln N}{\epsilon} L$.

We will use these bounds to transition to the bandit feedback context via the following two steps:

1. Create an unbiased estimator $\tilde{c}^t$, that is that $\mathbb{E}\left[\tilde{c}^t(a)\right] = c^t(a) \forall a \in \mathcal{A}$ such that $\tilde{c}^t(a) = 0 \ \forall a \neq a^t$.

2. Run a full information algorithm on $\tilde{c}^t$.

We use importance sampling to construct this unbiased estimator, described below.

$$\tilde{c}^t(a) = \begin{cases} \frac{c^t(a)}{p^t(a)} & \text{if } a = a^t; \\ 0 & \text{otherwise.} \end{cases}$$

Note this estimator satisfies $\mathbb{E}\left[\tilde{c}^t(a)\right] = \tilde{c}^t(a)$ and is thus unbiased.

Fitting to the online learning framework An important point is that the estimated costs depend on the action selected by the algorithm. This is not generally allowed by the online learning framework we described above as the adversary should select costs prior to when the learner draws her action from the distribution. However, when applying the full-information bound on the estimated costs, we will assume that the randomness that the algorithm is using is uncorrelated to the randomness of the partial-information algorithm (via which the estimated costs were determined). Intuitively, when applying the full-information bound, this is as if we ran the full-information algorithm on the estimated costs with fresh randomness. Therefore, it fits in the online learning framework we discussed before.

14.1.4 Bounds for Cost

$$\begin{aligned}
\mathbb{E}\left[\sum_t c^t(a^t)\right] &= \sum_t \sum_a p^t(a) \cdot c^t(a) \\
&= \sum_t \sum_a p^t(a) \cdot \mathbb{E}\left[\tilde{c}^t(a)\right] && \text{as } \tilde{c}^t \text{ is an unbiased estimator} \\
&= \mathbb{E}\left[\sum_t \sum_a p^t(a) \cdot \tilde{c}^t(a)\right] && \text{by linearity of expectations} \\
&\leq \mathbb{E}\left[(1+\epsilon)\sum_t \tilde{c}^t(a^*) + \frac{L \cdot \ln N}{\epsilon}\right] && \text{by the Low Approximate Regret property} \\
&= (1+\epsilon)\sum_t \mathbb{E}\left[\tilde{c}^t(a^*)\right] + \frac{L \cdot \ln N}{\epsilon} && \text{by linearity of expectations} \\
&\leq (1+\epsilon)\sum_t c^t a^* + \frac{L \cdot \ln N}{\epsilon} && \text{as } \tilde{c}^t \text{ is an unbiased estimator}
\end{aligned}$$

However the problem is that L can be unbounded since p^t can be arbitrarily close to 0.

To circumvent this issue, we set threshold γ and freeze actions a with $p^t(a) < \gamma$ (don't play them). Then renormalize to get a new probability distribution w^t.

$$w^t(a) = \begin{cases} 0 & \text{if } P^t(a) < \gamma; \\ \frac{p^t(a)}{1 - \sum_{a^t : p^t(a) < \gamma} p^t(a^t)} & \text{otherwise.} \end{cases}$$

Estimation costs:

$$\tilde{c}^t(a) = \begin{cases} \frac{c^t(a)}{w^t(a)} & \text{if } a = a^t; \\ 0 & \text{otherwise} \end{cases}$$

Now we make the following crucial observations:

- The maximum estimated cost that can be awarded is $L = \frac{1}{\gamma}$ as, if $p^t(a) < \gamma$, the estimated cost is 0. So L is bounded by this quantity.
- However, this estimator is not unbiased generally. We observe though that:

  It is unbiased for nonfrozen actions a, that is $\mathbb{E}\left[\tilde{c}^t(a)\right] = c^t(a)$ if $p^t(a) \geq \gamma$.

  It is biased in one direction $\forall a \in \mathcal{A}, \mathbb{E}\left[\tilde{c}^t(a)\right] \leq c^t(a)$.
- For all non-frozen arms a, the probability post freezing is upper bounded by

  $$w^t(a) \leq \frac{p^t(a)}{1 - \gamma N}.$$

  This is because the total probability in the full information algorithm of frozen arms should be less than γN as each frozen arm has probability less than γ.

14.1.5 Combining Pieces

$$\begin{aligned}
\mathbb{E}\left[\sum_t c^t(a^t)\right] &= \sum_t \sum_a w^t(a) c^t(a) \\
&= \sum_t \sum_a w^t(a) \mathbb{E}\left[\widetilde{c}^t(a)\right] && \text{as } \widetilde{c}^t \text{ is unbiased for non-frozen arms } (a : w^t(a) > 0). \\
&\leq \sum_t \sum_a \frac{p^t(a)}{1-\gamma N} \mathbb{E}\left[\widetilde{c}^t(a)\right] && \text{by the upper bound on the post-freezing probability} \\
&= \frac{1}{1-\gamma N} \mathbb{E}\left[\sum_t \sum_a p^t(a) \widetilde{c}^t(a)\right] && \text{by linearity of expectations} \\
&\leq \frac{1}{1-\gamma N} \mathbb{E}\left[(1+\epsilon) \sum_t \widetilde{c}^t(a^*) + \frac{L \cdot \ln N}{\epsilon}\right] && \text{by the Low Approximate Regret property} \\
&= \frac{1+\epsilon}{1-\gamma N} \sum_t \mathbb{E}\left[\widetilde{c}^t(a^*)\right] + \frac{L \cdot \ln N}{\epsilon} && \text{by linearity of expectations} \\
&\leq \frac{1+\epsilon}{1-\gamma N} \sum_t c^t(a^*) + \frac{L \cdot \ln N}{\epsilon} && \text{by the one-direction bias of the estimator} \\
&\leq \frac{1+\epsilon}{1-\gamma N} \sum_t c^t(a^*) + \frac{\ln N}{\gamma \cdot \epsilon} && \text{by the bound on } L
\end{aligned}$$

Setting $\gamma = \frac{\epsilon}{N}$, the above bound becomes:

$$\mathbb{E}\left[\sum_t c^t(a^t)\right] \leq \frac{1+\epsilon}{1-\epsilon} \sum_t c^t(a^*) + \frac{N \cdot \ln N}{\epsilon^2}.$$

Assuming $\epsilon < 1/3$, we can upper bound $\frac{1+\epsilon}{1-\epsilon} \leq 1 + 3\epsilon$. Applying this inequality and setting $\bar{\epsilon} = 3\epsilon$, we derive the Low Approximate Regret guarantee:

$$\mathbb{E}\left[\sum_t c^t(a^t)\right] \leq (1+\bar{\epsilon}) \sum_t c^t(a^*) + 9\frac{N \cdot \ln N}{\bar{\epsilon}^2}.$$

14.1.6 To sum up

To summarize, from any full information algorithm with Low Approximate Regret $\frac{\ln(N)}{\epsilon}$, we can derive a partial information algorithm that uses the above algorithm a a black box and has Low Approximate Regret $\frac{N \log(N)}{\epsilon^2}$. This is done via running the full information algorithms on some estimated costs and freezing arms with very small probability.

We note that, for specific full information algorithms, one can indeed obtain a linear dependence on ϵ on the denominator (instead of quadratic). In particular, this happens for Multiplicative Weights and for Online Mirror Descent with Log-Barrier regularizer.