COS-511: Learning Theory **Spring 2017**

Lecture 22: Learning with Partial Feedback

Lecturer: Roi Livni

22.1 Learning with Partial Feedback

Recall the "learning from expert advice" game: there are N experts indexed $i \in [N]$, and in the t-th round, the player picks one expert i_t and "suffers the loss" associated with that expert $f_t(i_t) \in \{0,1\}$. The player's goal is to choose the experts $i_1, \ldots, i_T$ so as to minimize the *regret*: the difference between the total loss suffered by the experts whom the player selected, and the total loss suffered by expert i^* who was, in hindsight, the best:

$$\text{Regret}_T = \sum_{t=1}^{T} f_t(i_t) - \min_{i^* \in [N]} \sum_{t=1}^{T} f_t(i^*)$$

Crucially, in each round t of the "learning from expert advice" game, the player is given not only the loss of the expert whom she chose, which is a single number $f_t(i_t) \in \{0,1\}$, but also the loss of *all the other experts*, which is a function $f_t : [N] \to \{0,1\}$. The multiplicative weights (MW) algorithm exploited this additional information, by maintaining a "weight" for each expert and penalizing *all* experts who were wrong in round t, not just the expert i_t who was chosen. However, in many real-world problems that we may wish to model using online convex optimization, the "player" does not have access to the "loss" associated with decisions other than the one that the player made. For example, if a (hypothetical) Princeton undergraduate is trying to optimize his course schedule over his $T = 8$ semesters in college so as to minimize the number of essays that he has to write, he does not know how many essays were assigned in classes that he did not take. This motivates the so-called **multi-armed bandit** problem, a variant of "learning from expert advice" in which the player only observes the regret of the expert whom she chose.

22.1.1 The Multi-Armed Bandit Problem

We have N experts. In each round of play $t = 1, 2, \ldots T$, the player picks one expert $i_t \in [N]$ and suffers a loss $f_t(i_t) \in [0,1]$. (Note that the loss here is real-valued, not binary as it was above.) The goal is to minimize regret, defined as above.

Recall that in "learning from expert advice," the multiplicative weights algorithm guaranteed a regret bounded by $\sqrt{T \log N}$. Is there an algorithm that can attain the same regret bound in the multi-armed bandit case? Unfortunately,

Theorem 22.1. *Any algorithm for the multi-armed bandit problem might attain $\Omega(N)$ regret in the worst case.*

Proof. Consider an situation where $N - 1$ of the experts always give a loss of 1, and one expert, chosen uniformly at random, always gives a loss of 0.

An optimal learner will keep trying experts until it finds the good one, thus:

$$\begin{aligned}
\mathbf{E}[\text{Regret}_T] &= \mathbf{E}\left[\sum_{t=1}^{T} f_t(i_t) - \min_{i^* \in [N]} \sum_{t=1}^{T} f_t(i^*)\right] \\
&= \mathbf{E}\left[\sum_{i=1}^{T} f_t(i_t)\right] \\
&= \mathbf{E}\left[\text{number of iterations to find good expert}\right] \\
&= \sum_{j=1}^{N} \mathbb{P}[\text{takes} \geq j \text{ iterations to find good expert }] \\
&= \frac{N-1}{N} + \frac{N-1}{N}\frac{N-2}{N-1} + \frac{N-3}{N-2}\frac{N-2}{N-1}\frac{N-1}{N} + \dots \\
&= \frac{N-1}{N} + \frac{N-2}{N} + \frac{N-3}{N} + \ldots + \frac{1}{N} \\
&= \Omega\left(\frac{N^2}{N}\right) \\
&= \Omega(N)
\end{aligned}$$

22.1.2 Exp3

Algorithm 1 EXP3 Algorithm

Let $\mathbf{x}_1 = \frac{1}{N}\mathbf{1}$
for t=1 to T **do**
  choose $i_t \sim \mathbf{x}_t$ and play i_t
  set $\hat{\ell}_t(i) = \begin{cases} \frac{1}{\mathbf{x}_t(i_t)} f_t(i) & \text{if } i = i_t \\ 0 & \text{otherwise} \end{cases}$
  update $\mathbf{y}_{t+1}(i) = \mathbf{x}_t(i)\, e^{-\epsilon \hat{\ell}_t(i)}$ $\quad$ $\mathbf{x}_{t+1} = \frac{\mathbf{y}_{t+1}}{||\mathbf{y}_{t+1}||_1}$
end for

The simple algorithm presented above for the multi-armed bandit problem can be improved upon if we do away with the distinction between "explore" and "exploit" steps.

Theorem 22.2. *The regret of EXP3, with* $\epsilon = \sqrt{\frac{\log n}{Tn}}$ *is bounded by* $O(\sqrt{N \log N}\sqrt{T})$.

Proof. First, we claim that for every i, $\mathbb{E}(\hat{\ell}_t(i)) = \mathbf{x}_t(i)\frac{\ell(i)}{\mathbf{x}_t(i)} + (1 - \mathbf{x}_t(i)) \cdot 0 = \mathbb{E}\left(\ell_t(i)\right)$. We also have that $\mathbb{E}(\mathbf{x}_t \cdot \hat{\ell}(i_t)) = \mathbb{E}(\ell_t(i_t))$. Next, we bound the second moment of $\hat{\ell}$:

$$\mathbb{E}\left(\mathbf{x}_t^\top \hat{\ell}_t^2\right) = \mathbb{E}\left(\mathbf{x}_t(i_t)\frac{\ell^2(i_t)}{\mathbf{x}_t(i_t)^2}\right) \leq \mathbb{E}\left(\frac{1}{\mathbf{x}_t(i_t)}\right) = n \tag{22.1}$$

We now exploit the regret bound for MW algorithm we derived in previous classes. Namely:

$$\sum_{t=1}^{T} \hat{\ell}_t(i_t) - \hat{\ell}_t(i) \leq \epsilon \sum_{t=1}^{T} \mathbf{x}_t \cdot \hat{\ell}_t^2 + \frac{\log n}{\epsilon} \tag{22.2}$$

□

Taken together we obtain the following regret bound

$$\begin{aligned}\text{Regret}_T &= \mathbb{E}\left(\sum_{t=1}^{T}\ell_t(i_t) - \sum_{t=1}^{T}\ell_t(i)\right) = \sum_{t=1}^{T}\mathbb{E}\left(\sum_{t=1}^{T}\hat{\ell}_t(i_t) - \sum_{t=1}^{T}\hat{\ell}_t(i)\right) \\ &\leq \mathbb{E}\left(\epsilon\sum \mathbf{x}_t\cdot\hat{\ell}_t^2\right) + \frac{T\log n}{\epsilon} \\ &\leq \epsilon n T + \frac{\log n}{\epsilon}\end{aligned}$$

Is this the best regret bound one can attain for the bandit problem? No. In fact, the minimax rate for expected regret is $O(\sqrt{NT})$. In words, this means the following:

Theorem 22.3. *For the bandit problem:*

- *There exists an adversary that forces any algorithm to incur expected regret at least* $\Omega(\sqrt{NT})$.
- *There exists an algorithm that incurs at most* $O(\sqrt{NT})$ *expected regret against any adversary.*