CS 598: Theoretical Machine Learning

Lecturer: Pranjal Awasthi
Scribe: Abhishek Bhrushundi

Lecture # 10
Oct 9th, 2017

1 Online decision making

Online decision making is a generalization of the setting we studied in the previous lecture. In particular, our goal is to design an algorithm that can pick an action or make a decision at every time step from set of actions/decisions so that the loss/regret of the algorithm is not much worse than that of the best action or decision on the sequence. Here are some scenarios that are encompassed by this framework:

1. Page replacement in caches
2. Portfolio management
3. Routing in maps

2 RMW for online decision making

The actions/decisions are typically referred to as *experts* in the literature. Suppose we have N experts. On any given day t, we choose an expert to use from among the N experts, say i_t, and get feedback from the *world* at the end of day t in terms of a loss vector $\vec{\ell_t} = (\ell_{1,t}, \ldots, \ell_{N,t})$, where $\ell_{i,t}$ for $i \in [N]$, is the loss we would have accrued had we used expert i on day t. Since we used expert i_t, the actual loss we accrue on day t is going to be $\ell_{i_t,t}$.

For any strategy A of picking experts, let the total loss after T days be defined as

$$L_A = \sum_{t=0}^{T-1} \ell_{i_t,t}.$$

On the other hand, let L_B be the total loss of the best expert, i.e.

$$L_B = \min_{i \in [N]} \sum_{t=0}^{T-1} \ell_{i,t}.$$

Then, as before, the quantity $\frac{L_A}{T} - \frac{L_B}{T}$ is known as the regret of the strategy/algorithm A.

Let us assume for the rest of this section that for all $t \geq 0$ and $i \in [N]$, we have that $\ell_{i,t} \in [0, \gamma]$. Consider the following generalization of the RMW algorithm.

Randomized Multiplicative Weights algorithm for online decision making

- At $t = 0$ the algorithm initializes all weights to 1, i.e. $\forall i \in [N]$, $w_{i,0} = 1$.
- On day t, the algorithm chooses expert i with probability $\propto w_{i,t}$.

- After receiving the loss vector $\vec{\ell}_t$ at the end of day t, the algorithm updates all the weights as follows:

$$w_{i,t+1} = (1-\epsilon)^{\frac{\ell_{i,t}}{\gamma}} \cdot w_{i,t+1}.$$

Here ϵ is chosen depending on how large T is. In particular, we will see that $\epsilon = 2\gamma\sqrt{\frac{\log N}{T}}$.

As before, we will now show that, in expectation, the RMW algorithm achieves zero regret as $T \to \infty$.

Theorem 1. *Let L_A be the loss of RMW at the end of day T, and let L_B be the loss accrued by the best expert. Then,*

$$\mathbb{E}\left[\frac{L_A}{T} - \frac{L_B}{T}\right] \leq 2\gamma \cdot \sqrt{\frac{\log N}{T}}.$$

It is clear that in order to get any nontrivial guarantee on the regret from the above theorem, we must run the algorithm/use the strategy for long enough. In particular, to achieve regret at most ϵ, we need that $T \geq 4\gamma^2 \frac{\log N}{\epsilon^2}$.

Proof. We will use the same proof strategy as before: define the sum of all weights as the potential function, i.e. $W_t = \sum_{i=1}^N w_{i,t}$.

Now suppose, the best expert on the loss sequence is $j \in [N]$. Then, based on the same idea as in the previous lecture, we have that

$$W_T \geq (1-\epsilon)^{\frac{\sum_{t=0}^{T-1} \ell_{j,t}}{\gamma}}. \tag{1}$$

As for the upper bound, we have that

$$\begin{aligned}
W_t &= \sum_{i=1}^N w_{i,t-1}(1-\epsilon)^{\frac{\ell_{i,t-1}}{\gamma}} \\
\implies W_t &\leq \sum_{i=1}^N w_{i,t-1}\left(1 - \frac{\epsilon \cdot \ell_{i,t-1}}{\gamma}\right) \\
&\leq W_{t-1}\left(1 - \sum_{i=1}^N \frac{\epsilon \cdot \ell_{i,t-1}}{\gamma \cdot W_{t-1}}\right) \\
&= W_{t-1}\left(1 - \frac{\epsilon}{\gamma} \cdot \mathbb{E}[L_A^{(t-1)}]\right).
\end{aligned}$$

Here $\mathbb{E}[L_A^{(t)}]$ denotes the expected loss accrued by A on day t.

Recalling that $W_0 = N$, we may thus conclude that

$$\begin{aligned}
W_T &\leq N\left(1 - \frac{\epsilon}{\gamma} \cdot \mathbb{E}[L_A^{(0)}]\right) \ldots \left(1 - \frac{\epsilon}{\gamma} \cdot \mathbb{E}[L_A^{(T-1)}]\right) \\
\implies W_T &\leq N e^{-\frac{\epsilon \cdot \mathbb{E}[L_A]}{\gamma}},
\end{aligned} \tag{2}$$

where the last inequality uses the fact that $(1-x) \leq e^{-x}$ and that $\mathbb{E}[L_A] = \sum_{t=0}^{T-1} \mathbb{E}[L_A^{(t)}]$.

Combining (1) and (2), we get

$$\begin{aligned}
&\frac{L_B}{\gamma}\log(1-\epsilon) \leq \log N - \frac{\epsilon \cdot \mathbb{E}[L_A]}{\gamma} \\
\implies &\frac{\epsilon \cdot \mathbb{E}[L_A]}{\gamma} \leq \log N + \frac{L_B}{\gamma}\log\frac{1}{1-\epsilon} \\
\implies &\frac{\mathbb{E}[L_A]}{\gamma} \leq \frac{\log N}{\epsilon} + \frac{L_B \cdot (1+\epsilon)}{\gamma} \\
\implies &\frac{\mathbb{E}[L_A] - L_B}{T} \leq \frac{\gamma \log N}{T\epsilon} + \frac{\epsilon L_B}{T}
\end{aligned}$$

Since $L_B \leq \gamma \cdot T$, we get

$$\frac{\mathbb{E}[L_A] - L_B}{T} \leq \frac{\gamma \log N}{T\epsilon} + \epsilon \cdot \gamma$$

We can minimize the RHS by setting $\epsilon = 2\gamma\sqrt{\frac{\log N}{T}}$. This gives us the desired bound. □

3 Two limiting assumptions

So far whatever we've discussed suffers from the following two limiting assumptions:

1. N is finite and small: The number of experts need not be finite, and even if N is finite it could be really huge making all the algorithms we've seen so far infeasible in some sense — we cannot hope to maintain weights for all the experts. For example, in the case of routing, the experts are basically the different paths in the graph, and the number of paths could be very large.

2. We get to see the losses for all the experts everyday: Up till now, we assumed that after we've picked an expert on a given day, we get to see the losses for all the experts. In many scenarios, all that we may see is the loss of the expert that was picked. Examples of such scenarios are online auctions, and search rankings.

The second limitation motivates the *Bandit* framework.

4 Bandit framework

In the Bandit framework, on day t, after picking an expert i_t, we only get to see the loss $\ell_{i_t,t}$ instead of seeing the entire loss vector $\vec{\ell}_t$. The natural question to ask is whether we can achieve zero regret in this setting. It turns out this is indeed possible, and is achieved via an algorithm known as the EXP3 algorithm.

The EXP3 algorithm achieves the following bound in the bandit setting, provided all the losses are in $[0, 1]$:

$$\mathbb{E}\left[\text{Regret}(\text{EXP3})\right] \leq \sqrt{\frac{N \log N}{T}},$$

where N is the number of experts, and T is the number of days.

Before we get into the EXP3 algorithm, we consider the following naive adaptation of the RMW algorithm to the bandit setting:

- Pick expert i_t on day t.
- At the end of day t, we receive the loss $\ell_{i_t,t}$ for the expert i_t. We can create the vector $\vec{\ell_t}$ by setting the loss of all the other experts to zero: $\vec{\ell_t} = (0, \ldots, \ell_{i_t}, \ldots, 0)$, and then use this loss vector in the RMW algorithm.

In general, this method has a serious bottleneck that we shall sketch below, although it can be formalized.
An adversary can work in two phases: in the first phase the adversary can drive down the weight of an expert i^* so that i^* is not chosen in the second phase (with high probability). In the second phase, the adversary can play in a way so that i^* is the best expert in the phase, but since i^* is not likely to be picked, the above algorithm will accrue non-zero regret.

In the next lecture we will see the details of the EXP3 algorithm which modifies the above naive strategy in order to achieve zero regret.

Resources

http://www.cs.cmu.edu/ avrim/Papers/survey.pdf.