IEOR 8100-001: Learning and Optimization for Sequential Decision Making 02/08/16

Lecture 6+7: Adversarial Bandits

Instructor: Shipra Agrawal Scribed by: Erik Waingarten, Michael Hamilton

So far, we have been talking about multi-armed bandits where the rewards are stochastic, generated independently and identically from a fixed unknown distribution for each arm. Today, we'll look at a different setup: adversarial rewards. Instead of there being a distribution for each arm, we assume there is a hidden sequence for each arm i, $r_{i,1}, ..., r_{i,T}$. We observe $r_{i,t}$ if we pull arm i at time t. We assume that $r_{i,t} \in [0, 1]$.

The adversary creates these sequences before the algorithm begins. Because our algorithms will be randomized, the adversary may know the algorithm we use, but not the random bits used by the algorithm. This kind of adversary is called an *oblivious adversary*.

Recall that in the stochastic case, we had defined regret as: $\sum_{t=1}^{T}(\mu^* - \mu_{I_t})$, where μ^* is the highest expected reward among all arms, and μ_{I_t} is the expected reward from the arm we pulled at time t. In this case, the regret was defined with respect to benchmark of "always pull the arm with the highest expected reward."

In the adversarial case, the best possible algorithm may want to pull the arm with the highest reward at every step; however, since the future rewards have no relationship to the previous rewards, what would we possibly learn? Instead, we relax the benchmark, and compete with the strategy that always picks a single arm. This lends to a different definition of regret:

$$R(T) = \left(\max_i \sum_{t=1}^{T} r_{i,t}\right) - \sum_{i=1}^{T} r_{I_t,t}$$

With this aim, we may hope to learn over time which is the best single arm. Define $S_{\max} := \max_i \sum_{t=1}^{T} r_{i,t}$ as the highest reward an algorithm picking a single arm at all time steps can achieve, and $S_{\text{alg}} := \sum_{i=1}^{T} r_{I_t,t}$ as the reward we get from our algorithm. then,

$$\mathbf{E}[R(T)] = S_{\max} - \mathbf{E}[S_{\text{alg}}]$$

Here, expectation is with respect to the randomness in the algorithm.

First, we present an algorithm with under full information setting. Then we will give a way to transform it for the bandit setting.

1 Full Information Setting

In this section, we consider the following full information version of the problem: at each time step t, we pick arm I_t. We get the reward r_{I_t}, just like in the adversarial bandit problem, but now, we also get to observe the rewards of all other arms. This is a well studied problem, known as *experts problem* or *online learning*.

The most popular algorithm for this problem is called the Hedge Algorithm [Freund, Schapire, STOC 94]. It is based on a popular paradigm of algorithms called *multiplicative weights update*.

1.1 Hedge Algorithm

The weights are simply accumulating the rewards of the arm.

$$w_{i,t} = \exp\left(\sum_{s=1}^{t} \epsilon r_{i,s}\right)$$

Algorithm 1: Hedge Algorithm for Adversarial Bandits with Full Information

Maintain a weight for each arm. Initially, $w_{1,0} = \cdots = w_{N,0} = 1$.
foreach $t = 1, 2, \ldots,$ **do**
  Play arm i with probability $p_{i,t} = \dfrac{w_{i,t-1}}{\sum_{j=1}^{N} w_{j,t-1}}$.
  Observe all rewards $r_{i,t}$ and update $w_{i,t} = w_{i,t-1} \exp(\epsilon r_{i,t})$ for all $i = 1, \ldots, N$.
end

The intuition behind the algorithm is that we are taking a *soft max*. As $\epsilon \to \infty$, we play the arm with the best past rewards. As $\epsilon \to 0$, we play each arm with equal probability.

Theorem 1. *Assume $r_{i,t} \in [0,1]$ for all i and t. Let $S_{alg} = \sum_{t=1}^{T} r_{I_t,t}$ be the observed reward and $S_{\max} = \max_i \sum r_{i,t}$ be the reward of the algorithm playing the best arm. Then*

$$\boldsymbol{E}[R(T)] = S_{\max} - \boldsymbol{E}[S_{alg}] \leq 2\epsilon S_{\max} + \frac{\log(N)}{\epsilon}$$

For any choice $\epsilon \in [0,1]$.

We know that $S_{\max} \leq T$, so if we let $\epsilon = \sqrt{\frac{\log N}{T}} \leq 1$, we get

$$\mathbf{E}[R(T)] \leq 3\sqrt{T \log N}$$

The crucial lemma needed for the proof of above theorem is the following:

Lemma 2. (Weight Update Lemma)
Let $W_t = \sum_{i=1}^{N} w_{i,t}$ and let $r_t = \sum_{i=1}^{N} p_{i,t} r_{i,t}$, where $p_{i,t}$ are the choice probabilities. Then:

$$\frac{W_t}{W_{t-1}} \leq \exp\{(\epsilon^2 + \epsilon) r_t\} \tag{1}$$

Proof. Note first

$$\begin{aligned}
W_t &= \sum_i w_{i,t} \\
&= \sum_i w_{i,t-1} e^{\epsilon r_{i,t}} \\
\implies \frac{W_t}{W_{t-1}} &= \frac{\sum_i w_{i,t-1} e^{\epsilon r_{i,t}}}{W_{t-1}} \\
&= \sum_i p_{i,t} e^{\epsilon r_{i,t}} \\
(\text{using } \epsilon r_{i,t} \leq 1) &\leq \sum_i p_{i,t} (1 + \epsilon r_{i,t} + \epsilon^2 r_{i,t}^2) \\
&= 1 + \epsilon \sum_i p_{i,t} r_{i,t} + \epsilon^2 \sum_i p_{i,t} r_{i,t}^2 \\
(\text{using } r_{i,t} \leq 1) &\leq 1 + \epsilon \sum_i p_{i,t} r_{i,t} + \epsilon^2 \sum_i p_{i,t} r_{i,t} \\
&\leq 1 + \epsilon r_t + \epsilon^2 r_t \\
&\leq \exp\left\{(\epsilon + \epsilon^2) r_t\right\}
\end{aligned}$$

□

Now we can prove Theorem 1. We multiply all our weights, at each time:

$$\frac{W_T}{W_0} = \prod_{t=1}^{T} \frac{W_t}{W_{t-1}} \tag{2}$$

$$\leq \exp((\epsilon^2 + \epsilon) \sum_{t=1}^{T} r_t) \tag{3}$$

$$\log(W_T/W_0) \leq (\epsilon^2 + \epsilon)\mathbf{E}[S_{alg}] \tag{4}$$

So now we want to lower bound

$$W_T = \sum_{i=1}^{N} w_{i,t} \tag{5}$$

$$= \sum_{i=1}^{N} \exp(\epsilon(r_{i,1} + \cdots + r_{i,T})) \tag{6}$$

$$\geq \max_{i \in [N]} \exp(\epsilon(r_{i,1} + \cdots + r_{i,T})) \tag{7}$$

$$= \exp(\epsilon S_{\max}) \tag{8}$$

$$\log W_T \geq \epsilon S_{\max} \tag{9}$$

And, we have $W_0 = N$. Substituting, we have

$$\epsilon S_{\max} - \log N \leq \log(W_T/W_1) \tag{10}$$

$$\leq \mathbf{E}[S_{alg}](\epsilon^2 + \epsilon) \tag{11}$$

$$S_{\max} - \frac{\log N}{\epsilon} \leq \mathbf{E}[S_{alg}](\epsilon + 1) \tag{12}$$

$$\frac{1}{(1+\epsilon)}(S_{\max} - \frac{\log N}{\epsilon}) \leq \mathbf{E}[S_{alg}] \tag{13}$$

$$(1 - 2\epsilon)(S_{\max} - \frac{\log N}{\epsilon}) \leq \mathbf{E}[S_{alg}] \tag{14}$$

$$\mathbf{E}[R(T)] \leq 2\epsilon\mathbf{E}[S_{\max}] + \frac{\log N}{\epsilon} \tag{15}$$

giving us the desired bound. Here, we used $\frac{1}{(1+\epsilon)} \geq (1 - 2\epsilon)$ for $\epsilon \in [0, 1]$.

2 Bandit Setting

2.1 Importance Sampling

We will adapt the Hedge algorithm from above to the bandit setting, where we don't see reward for every arm. We will use a general technique called *importance sampling*.

This general technique is useful when we can sample from one distribution $p(x)$, but we are interested in the expectation when we sample with respect to another distribution $q(x)$. The idea is simple: take the samples x and modify them by letting

$$\hat{x} = \frac{xq(x)}{p(x)}$$

So $\mathbf{E}_{x\sim p}[\hat{x}] = \mathbf{E}_{x\sim q}[x]$.

We apply the technique in the following way: at time t, we sample arm i with probability $p_{i,t}$. When we sample

arm i, we observe $r_{i,t}$, but not any other rewards. We use importance sampling to get an unbiased estimator for the reward of each arm.

We construct

$$\hat{r}_{i,t} = \begin{cases} \frac{r_{i,t}}{p_{i,t}} & I_t = i \\ 0 & I_t \neq i \end{cases}$$

This vector of rewards has the property that for each i and t,

$$\mathbf{E}[\hat{r}_{i,t}] = r_{i,t}.$$

This gives us an unbiased estimator for each arm. Now, we pretend we see every arm. The only issue now is that if $p_{i,t}$ is very small, we cannot bound the transformed reward. Fortunately, this issue has an easy fix, which is to always pull arm i with a non-negligible probability. This allows us to bound the reward, and therefore scale the regret.

EXP3 algorithm uses these estimators of rewards as a substitute for the rewards used in the Hedge algorithm.

2.2 EXP3 Algorithm [Auer, Cesa-Bianchi, Freund, Schapire, FOCS 95]

EXP3 algorithm as stated here uses two parameters γ, ϵ. We will provide the specific choices of these parameters in the regret analysis.

Algorithm 2: EXP3 Algorithm

Inputs $\epsilon \in [0,1], \gamma \in [0,1]$.
Maintain a weight for each arm. Initially, $w_{1,0} = \cdots = w_{N,0} = 1$.
foreach $t = 1, 2, \ldots,$ **do**
Play arm $I_t = i$ with probability

$$p_{i,t} = (1-\gamma)\frac{w_{i,t-1}}{\sum_{j=1}^{N} w_{j,t-1}} + \frac{\gamma}{N}.$$

Observe reward $r_{I_t,t}$.
Update $w_{i,t} = w_{i,t-1}\exp(\epsilon\hat{r}_{i,t})$. Here,

$$\hat{r}_{i,t} = \begin{cases} \frac{r_{i,t}}{p_{i,t}} & I_t = i \\ 0 & I_t \neq i \end{cases}$$

end

2.3 A simple but suboptimal analysis of EXP3

Note that we are following Hedge algorithm with probability $1-\gamma$, with $\hat{r}_{i,t}$ substituted for the reward of each arm. However, Hedge algorithm's analysis required $r_{i,t} \leq 1$. Note that since $p_{i,t} \geq \frac{\gamma}{N}$, we have $\hat{r}_{i,t} \leq \frac{N}{\gamma}$. Scaling the rewards to apply Hedge algorithm's analysis, we get the following regret bound for EXP3:

$$\mathbf{E}[R(T)] \leq (2\epsilon S_{\max} + \frac{\log N}{\epsilon})\frac{N}{\gamma} + \gamma T$$

Where the regret is scaled by $\frac{N}{\gamma}$ because our rewards $\hat{r}_{i,t}$ may be as large as $\frac{N}{\gamma}$, and the last term comes from the fact that with probability γ (when we don't follow Hedge algorithm), we may incur regret at most 1 in every time step.

We can choose ϵ and γ to minimize the terms. We choose $\epsilon = \frac{1}{\sqrt{NT\log(N)}}$ and $\gamma = \left(\frac{N}{T}\right)^{\frac{1}{4}}$, giving us $\mathbf{E}[R(T)] = O(N^{\frac{1}{4}}T^{\frac{3}{4}}\sqrt{\log(N)})$.

There is a lower bound of $\Omega(\sqrt{NT})$ which comes from the Stochastic model. The surprising result is that this regret bound is achievable up to log factors.

2.4 Near optimal regret bounds using EXP3

Theorem 3. *Assuming $r_{i,t} \in [0,1]$ for all i,t. Then, EXP3 Algorithm with $\gamma \in [0,1]$, $\epsilon = \frac{\gamma}{N}$, achieves regret bound,*

$$\boldsymbol{E}[R(T)] \leq 2N\epsilon S_{\max} + \frac{\log N}{\epsilon}$$

Substituting $\epsilon = \sqrt{\frac{\log N}{NT}}$ and $S_{\max} \leq T$, we get $3\sqrt{NT\log N}$ regret.

Lemma 4. (Weight Update Lemma v.2)
Let $W_t := \sum_{i=1}^{N} w_{i,t}$, $r_{i,t} \in [0,1]$ for all i,t, and $\hat{r}_{i,t}$ be as defined in the EXP3 algorithm. Then, for $0 \leq \gamma = N\epsilon \leq 1$,

$$\frac{W_t}{W_{t-1}} \leq \exp\left\{\frac{\epsilon r_{I_t,t} + \epsilon^2 \hat{r}_{I_t,t}}{1 - N\epsilon}\right\} \tag{16}$$

Proof. Note that since $p_{i,t} \geq \frac{\gamma}{N} = \epsilon$, we have $0 \leq \hat{r}_{i,t} \leq \frac{1}{\epsilon}$. We using this bound in calculations below.

$$\begin{aligned}
\frac{W_t}{W_{t-1}} = \frac{\sum_i w_{i,t-1} e^{\epsilon \hat{r}_{i,t}}}{W_{t-1}} &= \sum_i \frac{p_{i,t} - \epsilon}{1 - N\epsilon} e^{\epsilon \hat{r}_{i,t}} \\
\text{(using } \epsilon \hat{r}_{i,t} \leq 1 \text{...)} \quad &\leq \sum_i \frac{p_{i,t} - \epsilon}{1 - N\epsilon} \left(1 + \epsilon \hat{r}_{i,t} + \epsilon^2 \hat{r}_{i,t}^2\right) \\
&= 1 + \sum_i \frac{p_{i,t} - \epsilon}{1 - N\epsilon} \left(\epsilon \hat{r}_{i,t} + \epsilon^2 \hat{r}_{i,t}^2\right) \\
&\leq 1 + \epsilon \sum_i \frac{p_{i,t} \hat{r}_{i,t}}{1 - N\epsilon} + \epsilon^2 \sum_i \frac{p_{i,t} \hat{r}_{i,t}^2}{1 - N\epsilon} \\
&= 1 + \epsilon \frac{r_{I_t,t}}{1 - N\epsilon} + \epsilon^2 \frac{r_{I_t,t} \hat{r}_{I_t,t}}{1 - N\epsilon} \\
\text{(using } r_{I_t,t} \leq 1 \text{...)} \quad &\leq 1 + \frac{\epsilon}{1 - N\epsilon} r_{I_t,t} + \frac{\epsilon^2}{1 - N\epsilon} \hat{r}_{I_t,t} \\
&\leq \exp\{\frac{\epsilon r_{I_t,t} + \epsilon^2 \hat{r}_{I_t,t}}{1 - N\epsilon}\}
\end{aligned}$$

The last equality follows from noting:

$$p_{i,t}\hat{r}_{i,t} = \begin{cases} r_{i,t} & \text{if } I_t = \text{i} \\ 0 & \text{otherwise} \end{cases}$$

because of how we define the estimator. The last inequality follows again from $1 + x \leq e^x$. □

Now armed with this lemma we can tackle the analysis of EXP3.

Proof of Theorem 3. By taking the log of both sides of the above lemma:

$$\log\{\frac{W_T}{W_1}\} \leq \frac{\epsilon \sum_t r_{I_t,t} + \epsilon^2 \sum_t \hat{r}_{I_t,t}}{1 - N\epsilon} \tag{17}$$

$$= \frac{1}{1 - N\epsilon} \left(\epsilon S_{\text{alg}} + \epsilon^2 \sum_t \hat{r}_{I_t,t} \right) \tag{18}$$

Now note $W_t = \sum_i \exp\{\sum_t \epsilon \hat{r}_{i,t}\} \geq \exp\{\sum_t \epsilon \hat{r}_{j,t}\}$, for any fixed j, since all the terms are positive. This, along with $W_1 = N$ implies:

$$\log\{\frac{W_T}{W_1}\} \geq \epsilon S_{\max} - \log(N) \tag{19}$$

putting these two expressions together gives:

$$\epsilon S_{\max} - \log(N) \leq \frac{1}{1 - N\epsilon} \left(\epsilon S_{\text{alg}} + \epsilon^2 \sum_t \hat{r}_{I_t,t} \right) \tag{20}$$

Taking expectation over both sides gives:

$$\epsilon S_{\max} - \log(N) \leq \frac{1}{1 - N\epsilon} \left(\epsilon E[S_{\text{alg}}] + \epsilon^2 \sum_t \sum_i r_{i,t} \right) \tag{21}$$

where we've used that $E[\hat{r}_{I_t,t}] = \sum_i \frac{r_{i,t}}{p_{i,t}} Pr(I_t = i) = \sum_i r_{i,t}$.

Using $\sum_i \sum_t r_{i,t} \leq N S_{\max}$, and dividing by ϵ:

$$S_{\max} - \frac{\log(N)}{\epsilon} \leq \frac{1}{1 - N\epsilon} \left(E[S_{\text{alg}}] + \epsilon N S_{\max} \right) \tag{22}$$

rearranging terms gives:

$$E[S_{\text{alg}}] \geq (1 - N\epsilon) S_{\max} - N\epsilon S_{\max} - \frac{\log\{N\}}{\epsilon} \tag{23}$$

subtracting the above lower bound from $S_{\max}$ gives the result. □

Now that we have this theorem, lets play with it a little. Choosing $\epsilon = \sqrt{\frac{\log\{N\}}{S_{max} N}}$ gives regret

$$E[R(T)] \leq \sqrt{N S_{max} \log\{T\}} \tag{24}$$

which we could achieve if we knew $S_{\max}$, otherwise one may use an upper bound of T on $S_{\max}$ to get $\sqrt{NT \log(T)}$ regret. The above regret implies an $O(\sqrt{N})$ dependence on N. If the number of arms is quite large our regret can be very poor where as in the full information case we didn't have this dependence. This corresponds to the inherit need to explore the arms, in the case of where we don't observe non-pulled arms.

Extensions. [1] considers several extensions of the above algorithm. For example, an extension is to get a similar "with high probability" regret by slightly modifying the way we estimate $r_{i,t}$. Namely changing $\hat{r}$ to be:

$$\hat{r}_{i,t} = \frac{r_{i,t}}{p_{i,t}} + \frac{1}{p_{i,t}\sqrt{NT}} \tag{25}$$

which will give us regret of order $\sqrt{NT \log(\frac{N}{\delta})}$ $w.p.$ $1 - \delta$. Another extension is that if we don't know $S_{\max}$ or even T in advance, we can still adapt the algorithm to give regret $\approx \sqrt{N S_{\max} \log(N)}$.

3 Adaptive Adversary

In these previous sections we considered the comparison of our online algorithm versus an *oblivious* adversary. The adversary had to pick the sequence of rewards as inputs to the problem before the run of the algorithm begins. Here, we consider a more powerful adversary who observes the decisions of the algorithm, and modifies the sequence to based on the actions of the algorithm. The following definition captures this notion.

Definition 5. *We'll call an adversary* **adaptive** *if at time t, after the algorithm picks an arm I_t, the adversary picks $r_{I_t,t}$. The goal of the algorithm is again to maximize $\sum_t r_{I_t,t}$.*

Another more precise way of formulating this is to imagine a sequence of functions $\{f_i\}_{i=1}^T$ as fixed. Then the sequence of events is: At time t,

1. Adversary sets $r_{i,t} = f_t(I_1, \ldots, I_{t-1}, i)$ for all i
2. ALG picks an arm I_t

Then regret is

$$R(T) = \sum_t f_t(I_1^*, \ldots, I_t^*) - f_t(I_1, \ldots, I_t) \tag{26}$$

And, regret compared to the benchmark of fixed choice of arm at all time steps:

$$R(T) = \sum_t f_t(i, \ldots, i) - f_t(I_1, \ldots, I_t) \tag{27}$$

This adversary is quite strong. In fact it's not hard to see that no algorithm can achieve sub linear regret. To see why, note on the first pull there must be some arm who's choice probability in the algorithm is $p < 1$; call this arm i. In this case, consider the adversary who gives reward 0 for first pull. And, after the first pull, the adversary gives 0 reward if the first pull wasn't arm i and 1 otherwise. Then any constant sequence player that always plays arm $j \neq i$ gets reward T but any algorithm can only get pT which implies linear regret.

This motivates us to investigate weaker notions of adversary. There are two common ways we relax the opponent, the first is by capping the adversaries "memory", i.e. the number of previous pulls of the algorithm we allow the adversary to remember. Refer to [2] for further results in this setting. The second is via the notion of weak regret.

Definition 6. *We define* **Weak Regret** *as*

$$\sum_t f_t(I_1, \ldots, I_{t-1}, I_t^*) - f_t(I_1, \ldots, I_{t-1}, I_t) \tag{28}$$

Lets unpack this definition. Here the reward sequences are not fixed in advance, they can depend adaptively on the actions of the algorithm, but now regret is defined compared to the optimal algorithm evaluated on the path (action sequence) *generated by the algorithm.*

[1] provide regret bounds for adaptive adversary under this definition of weak regret.

References

[1] The Nonstochastic Multiarmed Bandit Problem, Peter Auer, Nicol Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. SIAM Journal on Computing, Volume 32 Issue 1, 2003 Pages 48 - 77.

[2] Online Bandit Learning against an Adaptive Adversary: from Regret to Policy Regret, Ofer Dekel, Ambuj Tewari, and Raman Arora. ICML 2012.