CS 7545: Machine Learning Theory **Fall 2018**

Lecture 15: Stochastic Multi-Armed Bandits

Lecturer: Jacob Abernethy *Scribes: Mohit Prabhushankar, Varun Gupta*

15.1 Recap

In the last lecture we analyzed EXP3 algorithm for the case of adversarial multi-armed bandits. In the multi-armed bandit setting, the algorithm pays out $l^t_{i_t}$, only for the action that was taken. In an adversarial setting, the unobserved pay out vector l^t is chosen arbitrarily but fixed in advance. In this lecture we introduce the setting of stochastic multi-armed bandits where the pay out vector l^t is drawn from an i.i.d distribution across time.

15.2 Stochastic Bandits

15.2.1 Setting

For t = 1 ... T

- Alg plays $i_t \in$ [K]
- Alg earns/observes $X^t_{i_t}$

where there are K possible choices for arms at time t and the chosen arm i pays $X^t_i \sim D_i$
Note that the distributions for any 2 arms, i and j, $D_i \neq D_j, \forall i \neq j$. Also, X^t_i is independent of $X^{t'}_i, \forall t \neq t'$. Hence the name stochastic setting.

In this setting, the best arm i is the one that minimizes $\mu_i = \underset{X \in D_i}{\mathbb{E}}[X]$. This is different from previous lecture's adversarial setting because of the stochastic nature of X.

The regret in the stochastic setting is defined as

$$\text{Reg}_t := \mu_{i^*} T - \sum_{t=1}^{T} X^t_{i_t} \tag{15.1}$$

where $i^* = \arg\max \mu_i$. The first term is deterministic while the second term is random. Hence an expectation needs to be taken across arms.

15.2.2 Warm-up Problem : Two coin problem

Consider two coins, the first a fair coin while the second a weighted coin. The distributions of the outputs of both the coins are shown below.

$$D_1 := \begin{cases} 0(\text{Heads}) & ,\text{with probability } 1/2 \\ 1(\text{Tails}) & ,\text{with probability } 1/2 \end{cases}$$

$$D_2 := \begin{cases} 0(\text{Heads}) & ,\text{with probability } 1/2 - \epsilon \\ 1(\text{Tails}) & ,\text{with probability } 1/2 + \epsilon \end{cases}$$

Algorithm chooses coin 1 or 2 in each round. Let T be the total number of coin tosses, N_1 be the total number of times coin 1 was played and N_2 be the total number of times coin 2 was played. We find the regret according to Eq.15.1

By inspection, the best coin, i.e the arm with highest probability of occurance, is coin 2. Therefore, $\mu_{i^*} = \left(\frac{1}{2} + \epsilon\right)$. Hence, the first term in Eq.15.1 is given by $T\left(\frac{1}{2} + \epsilon\right)$.

The second term over both the arms is,

$$\text{No. of times played on coin } 1 \times \mu_1 + (\text{T} - \text{No. of times played on coin } 1) \times \mu_2$$
$$\text{N}_1 \times \frac{1}{2} + (\text{T} - \text{N}_1)\left(\frac{1}{2} + \epsilon\right)$$

Combining both the terms, the regret is given by,

$$\text{Reg}_T = T\left(\frac{1}{2} + \epsilon\right) - \text{N}_1 \times \frac{1}{2} + (\text{T} - \text{N}_1)\left(\frac{1}{2} + \epsilon\right) \tag{15.2}$$
$$\text{Reg}_T = \text{N}_1\epsilon \tag{15.3}$$

Intuitively, the regret of the above problem is given by : How many times you played wrong arm $\times$ How much you lose.

15.3 ϵ-Greedy Algorithm

The above scenario assumed that the distributions D_1 and D_2 were provided. This is not always the case. We find the distribution by using the ϵ-greedy algorithm. The full algorithm is given below.

Algorithm 1: ϵ-Greedy Algorithm

---

for $t = 1, \ldots, m$ **do**
- Play arm 1
- $\hat{\mu_1} = \frac{1}{m}\sum_{t=1}^{m} X_1^t$

for $t = 1 + m, \ldots, 2m$ **do**
- Play arm 2
- $\hat{\mu_2} = \frac{1}{m}\sum_{t=m+1}^{2m} X_2^t$

For the remainder of game,

for $t = 2m + 1, \ldots, T$ **do**
- Play $i_t = \text{arg max } \hat{\mu_i}$

The first two stages from $t = 1, \ldots, 2m$ are exploration stages and the last stage until T, is called exploitation.

15.3.1 Regret Analysis

During exploration, we play the correct arm m times and wrong arm m times. Hence in Eq.15.1, the first term is,

$$\mu_{i^*}T = m\epsilon$$

The second term is given by (T $-2m$)$\times$ (The probability that arg max was wrong in the exploration stage).

$$\Pr(\text{arg max}_{wrong}) = \Pr\Big(\frac{1}{m}\sum_{t=1}^{m} X_1^t > \frac{1}{m}\sum_{t=m+1}^{2m} X_2^t\Big)$$

$$\Pr(\text{arg max}_{wrong}) = \Pr\Big(\sum_{t=1}^{m} X_1^t - \sum_{t=m+1}^{2m} X_2^t > 0\Big) \ldots \text{Taking out } \frac{1}{m}$$

In order to zero centre the random variables, we add $-\frac{m}{2}$ and $\Big(\frac{m}{2} + m\epsilon\Big)$ to both LHS and RHS to obtain,

$$\Pr(\text{arg max}_{wrong}) = \Pr\Big(\sum_{t=1}^{m}\Big(X_1^t - \frac{1}{2}\Big) - \sum_{t=m+1}^{2m}\Big(X_2^t - \frac{1}{2} - \epsilon\Big) > m\epsilon\Big)$$

Let $\Big(X_1^t - \frac{1}{2}\Big)$ be z_1 and $\Big(X_2^t - \frac{1}{2} - \epsilon\Big)$ be z_2. The expectations of both these RVs is 0 and their range width is 1. Therefore,

$$\Pr\Big(\sum_{t=1}^{2m} z_t > m\epsilon\Big)$$

$$\Pr\Big(\sum_{t=1}^{2m} z_t > m\epsilon\Big) \leq exp\Big(-2 \times \frac{(m\epsilon)^2}{2m}\Big) \ldots \text{By Hoeffding's Inequality}$$

$$= exp\Big(-m\epsilon^2\Big)$$

By setting $m = \frac{log(1/\delta)}{\epsilon^2}$, the RHS reduces to δ. Hence second term in Eq.15.1 is (T $-2m$)δ. Combining both the first and second terms, the expected regret is given by,

$$\mathbb{E}[\text{Reg}_T] \leq m\epsilon + (\text{T} - 2m)\delta$$

$$\mathbb{E}[\text{Reg}_T] \leq \frac{log(1/\delta)}{\epsilon^2}\epsilon + (\text{T} - 2m)\delta \ldots \text{Substituting for } m$$

$$\mathbb{E}[\text{Reg}_T] \leq \frac{log\ \text{T}}{\epsilon} + 1 \ldots \text{Taking } \delta = \frac{1}{T} \text{ and ignoring the -}2m \text{ term}$$

Because of the $\frac{1}{\epsilon}$ factor in the denominator, this algorithm works well only for a large value of ϵ. If ϵ gets smaller, $\mathbb{E}$[Reg] becomes bigger. Usually, it takes $\frac{1}{\epsilon^2}$ coin tosses to obtain good distributions. This is because $m = \frac{log(1/\delta)}{\epsilon^2}$ is $\mathcal{O}\Big(\frac{1}{\epsilon^2}\Big)$.

15.4 The Upper Confidence Bound Algorithm

There are two limitations to the $\epsilon-$greedy algorithm:

- requires that we know ϵ in advance.
- only good for two arms.

What can we do in a more general setting, either when we don't have ϵ or have more than two arms? We will outline the Upper Confidence Bound (UCB) Algorithm (2002, Auer, Cesa-Bianchi, and Fischer), which we will analyze in more detail in the next lecture.

Setting

- We have k arms and each arm i has distribution $D_i \in \Delta_{[0,1]}$ with mean μ_i
- At each timestep t, the algorithm plays an arm $i^t \in [k]$ and recieves payoff $X_i^t \sim D_i$
- We define the best arm i^* as $i^* = \arg\max_{i \in [k]} \mu_i$
- We define expected regret at time T as

$$\mathbb{E}[\text{Regret}_T] := \mathbb{E}_{\text{arms, alg}}\Big[\sum_{t=1}^{T}(\mu_{i^*} - \mu_{i^t})\Big]$$

Algorithm 2: Upper Confidence Bound

for $t = 1, \ldots, k$ **do**
  pull $i^t = t$
for $t > k$ **do**
  $N_i^t = \sum_{s=1}^{t-1} 1_{[i_s = i]}$
  $\hat{\mu}_i^t = \sum_{s=1}^{t-1} \frac{x_i^s 1_{[i_s=i]}}{N_i^t}$
  $i^t = \arg\max_{i \in K} \hat{\mu}_i^t + \sqrt{\frac{\log(T)}{2N_i^t}}$

The second term $\frac{\log(T)}{N_i^t}$ is called the "exploration bonus," which incentivizes us to play unexplored arms. As N_i^t increases, the term goes to 0, and thus the incentive decreases as our confidence interval for arm i narrows.

In the next lecture, we will show that the UCB algorithm achieves

$$\mathbb{E}[\text{Regret}_T] \leq d \sum_{i \neq i^*} \frac{\log(T)}{\Delta_i},$$

where $\Delta_i = \mu_i^* - \mu_i$.