DS 102 Data, Inference, and Decisions **Fall 2019**

Lecture 16: Multi-Armed Bandits

Lecturer: Michael I. Jordan

1 Decisions and Learning

In our first lectures on FDR control, we talked about making sets of decisions. In particular, in online FDR control, we talked about making sets of decisions over time. However, in such a setting, the individual decisions could be completely unrelated. Starting today, we will be considering problems in which we make the same decision *repeatedly*, and we want to get better at making that decision. In other words, we want to *learn* the best decision. Learning involves a trade-off between *exploration* and *exploitation*: exploration refers to the idea of trying out new decisions with the goal of testing if they are better than the current best, while exploitation refers to repeating what currently seems the best decisions, without worrying about potentially unexplored decisions.

2 Exploration and Exploitation

We assume that we obtain *rewards* based on our decisions. Good decisions correspond to high rewards, while bad decisions correspond to low rewards. (Sometimes we use losses in this formalism instead of rewards, in which case, as expected, lower is better.)

The goal is to obtain a high rate of reward over time, and in doing so we aim to exploit our knowledge as it accrues. However, initially we have no prior knowledge on which decisions yield high rewards. As a result, we have to explore different decisions to figure that out. So, we might be willing to sacrifice a few bad decisions at the beginning in order to ensure good decisions in the future.

Too little exploration means that the learner may not discover which choice yields the highest reward. On the other hand, too much exploration means forgoing the opportunity to reap immediate high rewards in the possibly vain hope of future, even higher, rewards. So, at any given moment there is a trade-off - one has to decide whether they should exploit the knowledge they currently have, or explore further to improve their knowledge. Multi-armed bandits formalize this trade-off in the language of mathematics, and make this theory actionable.

3 The Multi-Armed Bandit

We consider a decision-maker who is given K options to choose from. We refer to these options as *arms*. Associated with each arm is a probability distribution over rewards. Initially, this distribution is unknown to the decision-maker. The decision-maker chooses an arm, usually referred to as *pulling* an arm, and receives a reward sampled from the corresponding reward distribution. This process is repeated over and over again.

It makes sense for the learner to pull each of the K arms at least once - any arm that is not pulled could indeed be far better than all other arms, in which case the learner could be missing out. One reasonable strategy is to pull each arm some number of times, analyze the gathered data to see which arm seems to have the highest average reward, and then keep pulling this arm going forward. This strategy is called *explore-then-commit*.

More precisely, the goal is to maximize the sum of rewards over all time steps. Sometimes future rewards are discounted, when rewards in the future are viewed as less valuable than more immediate rewards.

A typical way to measure the quality of realized decisions is via *regret*, which is the difference between our total, realized reward, and the reward of some kind of oracle who knows more than we do. Usually, this is an oracle who knows in advance which is the optimal arm.

There are different assumptions of the reward distributions in the literature. The setup described above, in which every arm has a fixed corresponding distribution, is typically referred to as a *stochastic environment*. Another type of environment is *adversarial*, in which the rewards are chosen adversarially for the learner, and not from a fixed distribution. There are also *contextual bandits*, which have multiple ”contexts”, and a set of arms for each context. Another type of bandits are *Markovian bandits*, in which the context behaves according to a Markov chain. There are also *structured bandits*, in which the actions and the rewards have mathematical structure that can be exploited. In such settings, the feedback can be more detailed than simply the reward associated with the selected arm.

To see an example of structured bandits, consider the following example of path planning.

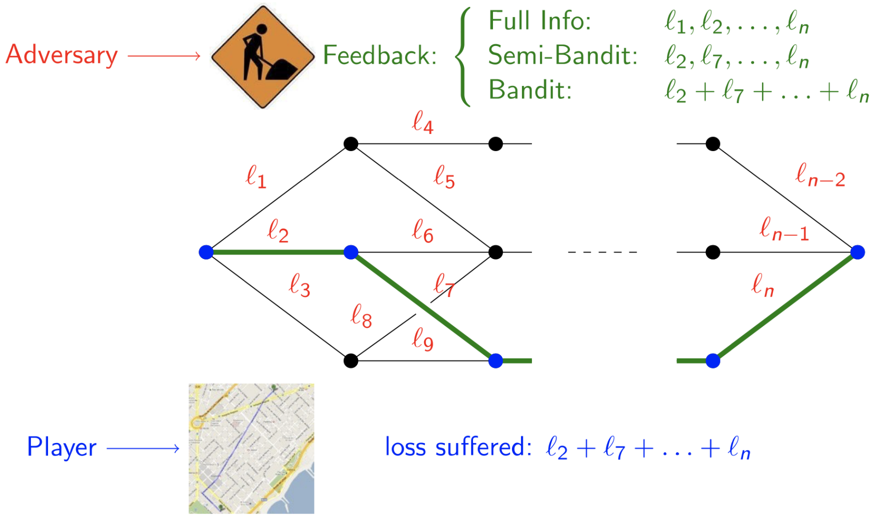
Adversary
Feedback:
Full Info: $\ell_1, \ell_2, \ldots, \ell_n$
Semi-Bandit: $\ell_2, \ell_7, \ldots, \ell_n$
Bandit: $\ell_2 + \ell_7 + \ldots + \ell_n$
ℓ_4
ℓ_1
ℓ_5
ℓ_{n-2}
ℓ_2
ℓ_6
ℓ_{n-1}
ℓ_3
ℓ_7
ℓ_n
ℓ_8
ℓ_9
Player
loss suffered: $\ell_2 + \ell_7 + \ldots + \ell_n$

Figure 1.1: Path planning.

The goal is to get from the leftmost point to the rightmost point, with minimum loss incurred. Each segment of the path, which corresponds to an edge in the above graph, corresponds to some loss ℓ_i. If we don't know the values of ℓ_i a priori, we could model this as a bandit problem, in which case we want to try out different paths and minimize the loss incurred by taking a path. In this case, arms correspond to different paths. If we get to see the losses as we go along, after crossing each edge, then we model this as a semi-bandit problem. In this case, we can associate losses with each segment of the road, which allows for more efficient learning, because in the full bandit setting we only observe the total loss once we take a whole path. Finally, we could also have full information available before having to explore different paths. In that case, we model the path planning problem as a standard supervised optimization problem, where we want to find the path minimizes the overall loss along that path.

Some real-world problems that to this day make decisions using the bandits framework: drug discovery, A/B testing, routing in networks, robot skills, human-computer interactions...

4 Mathematical Setup

We now introduce the formal mathematical setup of multi-armed bandits.

Let $\mathcal{A}$ denote a set of K arms. We will denote the reward distribution for arms $a \in \mathcal{A}$ by P_a.

By A_t we will denote the selected arm at time t, and by X_t the reward at time t, which is a random sample from distribution P_{A_t}.

Let n denote the total number of rounds. Then, our total reward is equal to

$$\sum_{t=1}^{n} X_t.$$

The goal is to find the arm a which has the highest corresponding mean of distribution P_a. Informally, we will refer to this mean as "the mean of arm a", and we will denote it by

$$\mu_a := \mathbb{E}_{Z \sim P_a}[Z].$$

The highest mean will be denoted by $\mu^* := \max_a \mu_a$. The arm with the highest mean (that is μ^*) will be denoted $a^* := \arg\max_a \mu_a$.

We formally define regret as:

$$R_n = n\mu^* - \mathbb{E}[\sum_{t=1}^{n} X_t].$$

In words, this is the difference between the best possible reward we could get if we knew which arm was the best one, and the expected regret we actually incur.

We also define the suboptimality gap for arm a as:

$$\Delta_a = \mu^* - \mu_a.$$

This is the difference between the mean of the best arm, and a fixed arm a.

We let $T_a(t)$ denote the number of times arm a is selected, before time t:

$$T_a(t) = \sum_{s=1}^{t} \mathbf{1}\{A_s = a\}.$$

Finally, we define the average reward observed for arm a by time t:

$$\hat{\mu}_{a,T_a(t)} = \frac{1}{T_a(t)} \sum_{s=1}^{t} X_s \mathbf{1}\{A_s = a\}.$$

5 UCB Algorithm

We describe one exploration-exploitation strategy called the UCB (Upper Confidence Bound) algorithm, and prove that the regret of this algorithm is logarithmic. Note that getting linear regret is trivial: if we keep picking any wrong arm forever, at each time step we will earn some constant amount of regret (Δ_a, for some a), which accumulates linearly over multiple time steps. Therefore, any sublinear regret is non-trivial. Logarithmic is quite remarkable however, as this is a very slowly growing function. This means that almost always we will pull exactly the best arm.

We assume the rewards are bounded, and, for simplicity, we take them to be in $[0, 1]$. The same analysis would hold for Gaussian rewards.

The UCB algorithm is defined as follows: at time $t \in \mathbb{N}$, it selects an arm according to:

$$A_t = \begin{cases} \text{argmax}_i \left(\hat{\mu}_i(t-1) + \sqrt{\frac{3\log(t)}{2T_i(t-1)}} \right), & \text{if } t > K \\ t, & \text{otherwise} \end{cases}.$$

Now we prove that this strategy achieves logarithmic regret. The proof is based on constructing confidence intervals using the Hoeffding concentration inequality.

In particular, due to boundedness of rewards, we can apply Hoeffding's inequality to get:

$$\mathbb{P}(\hat{\mu}_{a,t} - \mu_a > \epsilon) \leq \exp(-2t\epsilon^2).$$

This property will be crucial in our proof.

We state a simple auxiliary result, which says:

$$R_n = \sum_a \Delta_a \mathbb{E}[T_a(n)].$$

This follows simply by definition of Δ_a and R_n.

Now we claim that if $A_t = a$, then at least one of the following three events must be true:

$$\begin{aligned}
&\hat{\mu}_{i^*,T_{a^*}(t-1)} + \sqrt{\tfrac{3\log t}{2T_{a^*}(t-1)}} \leq \mu^* \quad (E_1) \\
&\hat{\mu}_{a,T_a(t-1)} > \mu_a + \sqrt{\tfrac{3\log t}{2T_a(t-1)}} \quad (E_2) \qquad . \\
&T_a(t-1) < \tfrac{6}{\Delta_a^2}\log(t) \quad (E_3)
\end{aligned}$$

Suppose E_1, E_2 and E_3 are all false, then we must have

$$\begin{aligned}
\hat{\mu}_{a^*,T_{a^*}(t-1)} + \sqrt{\frac{3\log t}{2T_{a^*}(t-1)}} &> \mu^* \\
&= \mu_a + \Delta_a \\
&\geq \mu_a + 2\sqrt{\frac{3\log n}{2T_a(t-1)}} \qquad . \\
&\geq \hat{\mu}_{a,T_a(t-1)} + \sqrt{\frac{3\log n}{2T_a(t-1)}}
\end{aligned}$$

This implies $A_t \neq a$, a contradiction.

Denote $u := \lceil \frac{6}{\Delta_a^2}\log(n) \rceil$. Then:

$$\begin{aligned}
\mathbb{E}\left[T_a(n)\right] &= \mathbb{E}\sum_{t=1}^{n} \mathbf{1}\left\{A_t = a \text{ and } E_3 \text{ is true at time } t\right\} + \mathbf{1}\left\{A_t = a \text{ and } E_3 \text{ is false at time } t\right\} \\
&\leq u + \mathbb{E}\sum_{t=u+1}^{n} \mathbf{1}\left\{A_t = a \text{ and } E_3 \text{ is false at time } t\right\} \\
&\leq u + \mathbb{E}\sum_{t=u+1}^{n} \mathbf{1}\{E_1 \text{ or } E_2 \text{ is true at time } t\} \\
&\leq u + \sum_{t=u+1}^{n} \mathbb{P}(E_1 \text{ is true at time } t) + \mathbb{P}(E_2 \text{ is true at time } t).
\end{aligned}$$

We now upper bound $\mathbb{P}(E_1 \text{ is true at time } t)$ by a union bound over all possible values of $T_a(t-1)$:

$$\begin{aligned}
\mathbb{P}(E_1 \text{ is true at time } t) &\leq \mathbb{P}\left(\exists s \in \{1, \cdots, t\} : \hat{\mu}_{a^*,s} + \sqrt{\frac{3\log t}{2s}} \leq \mu^*\right) \\
&\leq \sum_{s=1}^{t} \mathbb{P}\left(\hat{\mu}_{a^*,s} + \sqrt{\frac{3\log t}{2s}} \leq \mu^*\right) \\
&\leq \sum_{s=1}^{t} \frac{1}{t^3} \\
&= \frac{1}{t^2},
\end{aligned}$$

where we apply Hoeffding's inequality. Similarly we have:

$$\mathbb{P}(E_2 \text{ is true at time } t) \leq \frac{1}{t^2}.$$

Because $\sum_{t=u+1}^{n} \frac{1}{t^2} \leq \frac{1}{n} - \frac{1}{u} \leq \frac{1}{n}$, we can conclude:

$$\mathbb{E}\left[T_a(n)\right] \leq \lceil \frac{6}{\Delta_a^2} \log(n) \rceil + \frac{2}{n} \leq \frac{6}{\Delta_a^2} \log(n) + 3.$$

Multiplying by Δ_a and summing over a gives:

$$R_n \leq \sum_a \left(\frac{6}{\Delta_a} \log(n) + 3\Delta_a \right).$$