IEOR 8100-001: Learning and Optimization for Sequential Decision Making 01/25/16

Lecture 2: Stochastic Multi-armed bandit (IID model)

Instructor: Shipra Agrawal Scribed by: Antoine Désir

1 Problem definition

We consider a multi-armed bandit problem. There are N arms. At every (discrete) time step $t = 1, 2, 3, \ldots$, we pull one out of the N arms. Let $I_t \in \{1, \ldots, N\}$ be the arm pulled at the t^{th} time step. After pulling arm I_t, we observe a reward $r_t \in [0, 1]$. For a fixed time horizon T, the goal is to maximize the total reward

$$\text{maximize} \sum_{t=1}^{T} r_t. \tag{1}$$

1.1 Reward model

The next step in defining the model is to describe how the reward are generated. This is where the stochastic assumption and IID model comes in. In particular, we assume that the reward from each arm i follows a distribution ν_i with mean μ_i. When pulling an arm i, the reward will be generated independently from the distribution ν_i. More precisely, let H_{t-1} denote the history until time $t-1$ (including $t-1$). We can write

$$H_{t-1} = \{(I_1, r_1), \ldots, (I_{t-1}, r_{t-1})\}.$$

Then, our assumption on the reward can be written as

$$r_t | (H_{t-1}, I_t = i) \sim \nu_i$$

which also implies

$$\mathbb{E}[r_t | H_{t-1}, I_t = i] = \mu_i.$$

In other words, given the history up to time $t-1$ and the choice of arm I_t, the reward is drawn independently with respect to the distribution of the chosen arm.

1.2 Regret

We measure the performance of any algorithm as a function of how well it performs compared to a target algorithm which knows and chooses the best arm all the time. In particular, let $I^* = \text{argmax}_{i=1,\ldots,N} \mu_i$ and $\mu^* = \mu_{I^*}$ be the index of the best arm and the associated highest expected reward. We define the regret at time T by

$$R(T) = \sum_{t=1}^{T} (\mu^* - \mu_{I_t}). \tag{2}$$

Note that the goal is now to minimize $R(T)$. The choice of this regret as objective is motivated by several observations. First, note that this objective function uses the expected rewards. This is different from the expression (1) which was equal to the reward of a particular realization of an algorithm. Note that in the same spirit as (1),

we could have chosen the regret to be

$$\sum_{t=1}^{T}(s_t - r_t),$$

where s_t is the reward obtained at time t if we pull arm I^*. However, note that according to our model, at every time t, we only observe the reward of the arm we pull. Therefore, in general, s_t is not observable and therefore the objective is not very well defined. A hybrid objective could be

$$\sum_{t=1}^{T}(\mu^* - r_t).$$

This is a possible objective function, and it turns out that this is close to (2) for bounded rewards.

Finally, note that although $R(T)$ is defined in (2) using the expected rewards μ_i, it is itself a random variable. This is due to the fact that the choice of arm I_t is random since the observations are random. We can therefore define the expected regret as

$$\mathbb{E}[R(T)] = \sum_{t=1}^{T}(\mu^* - \mathbb{E}[\mu_{I_t}]). \tag{3}$$

2 Lower bound

2.1 A counting formula

For all arm $i = 1, \ldots, N$, let $\Delta_i = \mu^* - \mu_i$ and let $n_{i,t}$ be the number of times arm i was pulled until (and including) time t for all time steps $t = 1, \ldots, T$. We can rewrite the regret in time T as

$$R(T) = \sum_{t=1}^{T}(\mu^* - \mu_{I_t}) = \sum_{i=1}^{N}\sum_{t:I_t=i}(\mu^* - \mu_i) = \sum_{i=1}^{N} n_{i,T}\Delta_i = \sum_{i:\mu^*>\mu_i} n_{i,T}\Delta_i. \tag{4}$$

Rewriting the regret in this form hints at an adaptive algorithm whose goal is to quickly learn the arms with large Δ_is and discard them. Indeed, for arms where Δ_i is large, pulling them even a few times is very costly and incurs a higher regret.

2.2 Aside: Big-Oh notations

Definition 1.

$$\begin{aligned} f(n) = O(g(n)) &\iff \exists c > 0, \exists n_0, \forall n \geq n_0, f(n) \leq cg(n) \\ f(n) = o(g(n)) &\iff \forall c > 0, \exists n_0, \forall n \geq n_0, f(n) < cg(n). \end{aligned}$$

Example.

$$\begin{array}{lll} n = O(2n-5) & 2n = O(n) & n = O(n^2 - 10n) \\ n \neq o(2n-5) & n = o(n^2) & \end{array}$$

Definition 2.

$$\begin{aligned} f(n) = \Omega(g(n)) &\iff \exists c > 0, \exists n_0, \forall n \geq n_0, f(n) \geq cg(n) \\ f(n) = \omega(g(n)) &\iff \forall c > 0, \exists n_0, \forall n \geq n_0, f(n) > cg(n). \end{aligned}$$

2.3 A lower bound

In this section, we give a lower bound on the achievable regret of any algorithm. This gives an idea of what we can hope for in designing an algorithm. Here, we provide the lower bound is for a special case of Bernoulli reward distributions. Bernoulli distribution is a single parameter distribution, specified by the probability of success. An instance of Bernoulli multi-armed bandit problem is therefore entirely specified by $\Theta = (\mu_1, \ldots, \mu_N)$, where μ_i is the expected reward for arm i. More precisely, at each time t,

$$r_t = \begin{cases} 1 & \text{w.p. } \mu_{I_t} \\ 0 & \text{otherwise} \end{cases}.$$

For a given instance Θ, let $\mathbb{E}[R(T, \Theta)]$ be the expected regret of an online algorithm. We have the following result.

Theorem 3 (Lai and Robbins 1985 [3])**.** *Consider an online algorithm whose regret satisfies, for all fixed* Θ*,*

$$\mathbb{E}[R(T, \Theta)] = o(T^a), \forall a > 0. \tag{5}$$

Then, for every instance $\Theta = (\mu_1, \ldots, \mu_N)$ *such that* μ_i *are not all equal,*

$$\lim_{T \to \infty} \frac{\mathbb{E}[R(T, \Theta)]}{\ln(T)} \geq \sum_{i:\mu^* > \mu_i} \frac{\mu^* - \mu_i}{KL(\mu_i, \mu^*)}, \tag{6}$$

where $KL(,)$ *denote the Kullback-Leibler divergence, i.e. for all* i

$$KL(\mu_i, \mu^*) = \mu_i \ln\left(\frac{\mu_i}{\mu^*}\right) + (1 - \mu_i) \ln\left(\frac{1 - \mu_i}{1 - \mu^*}\right). \tag{7}$$

Informally, this theorem says that any good (also called consistent) algorithm, i.e. which makes less than sublinear number of mistakes on all instances, must make at least a logarithmic number of mistakes on every instance. Therefore, an online algorithm that can achieve a logarithmic regret is essentially an optimal online algorithm.

To see the necessity for a condition like (5), consider an algorithm that deterministically always pulls arm 1. Now, consider any instance Θ where arm 1 is indeed the best arm. Clearly, for this instance this algorithm will make no mistake, and therefore the regret cannot be lower bounded as in (6) for such instances. However, note that such an algorithm will have linear regret for instances where arm 1 is not the best arm (and not all μ_i are equal), and therefore, is not a "consistent algorithm".

This bound is extended in [3] for all single parameter distribution, and then for more general distributions in [1].

2.4 Examining the lower bound

Note that the lower bound is instance specific. Therefore, the performance of any algorithm is determined by the similarity between the optimal arm and other arms. In order to get a better intuition, we examine the Kullback-Leibler divergence. This function is used to measure the distance between two distributions. It is equal to 0 when the 2 distributions are equal and non-negative otherwise. However, unlike a proper measure, it is not symmetric. Also, note that the KL divergence is defined on distributions. Here, since the Bernoulli distribution is a single parameter distribution, we use its mean to represent the whole distribution. A better notation would have been $KL(\nu_i, \nu^*)$.

For Bernoulli distributions, we have the following inequality.

Lemma 4. *For all i,*

$$2\Delta_i^2 \leq KL(\mu_i, \mu^*) \leq \frac{\Delta_i^2}{\mu^*(1-\mu^*)}.$$

Those inequalities follow from standard inequalities (left hand side follow from Pinsker's inequality and the right hand side from $\ln(x) \leq x - 1$). Intuitively, these inequalities tell us that $KL(\mu_i, \mu^*) \sim \Delta_i^2$. Therefore, an algorithm achieving $\mathbb{E}[R(T, \Theta)] \leq \sum_{i \neq I^*} \frac{\ln(T)}{\Delta_i}$ for all T large enough, is close to optimal. Also, using (4), we see that a near-optimal algorithm should satisfy for all T, and all suboptimal arms i,

$$n_{i,T} \simeq \frac{\ln(T)}{\Delta_i^2}.$$

Finally, note that the number of times an optimal algorithm pulls each suboptimal arm grows with T. In other words, an optimal algorithm never stops exploring.

3 Algorithms

3.1 Empirical mean

A natural idea in designing an algorithm is to use the sample mean as a proxy for the real expected rewards. For every arm i and time s, let

$$\hat{\mu}_{i,s} = \frac{\sum_{t \leq s, I_t = i} r_t}{n_{i,s}}$$

be the sample mean of arm i at time step s. Since we will be using this estimate for our algorithms, we want to know how does this sample mean differs from the real mean. This can be quantified using Chernoff-Hoeffding bounds. Recall that we have assumed the rewards to be bounded (i.e. $r_t \in [0, 1]$).

Theorem 5 (Chernoff-Hoeffding bound)**.** *Let $X_1, \ldots, X_n$ be iid random variables in $[0, 1]$ such that for all i, $\mathbb{E}[X_i] = \mu$ and let $\bar{X}_n = (\sum_{i=1}^n X_i)/n$ be their sample mean. Then,*

$$\mathbb{P}(|\bar{X}_n - \mu| \geq \delta) \leq 2e^{-2n\delta^2}.$$

Since $\hat{\mu}_{i,t}$ is average of $n_{i,t}$ iid samples from distribution ν_i, mean μ_i, above bounds provide that at any time t,

$$|\hat{\mu}_{i,t} - \mu| \leq \sqrt{\frac{\ln(t)}{n_{i,t}}} \tag{8}$$

with probability at least $1 - \frac{2}{t^2}$.

3.2 Greedy Algorithm

The first natural candidate algorithm is one which pulls the best arm at every time step (after some fixed amount of exploration). Algorithm 1 details this algorithm.

Algorithm 1 Greedy Algorithm

1: For $t \leq cN$, select a random arm with probability $1/N$ and pull it.
2: For $t > cN$, pull arm with highest estimate, i.e. $I_t = \text{argmax}_{i=1,\ldots,N} \ \hat{\mu}_{i,t}$.

Here c is a constant.

The regret of this algorithm is linear. Indeed, since the exploitation phase has a fixed length, there is always a constant probability of not choosing the real best arm. Therefore, with constant probability, the algorithm incurs a linear regret which implies a linear regret for the algorithm. More formally, consider an example with 2 arms. Arm 1 has a fixed reward of $1/4$ and arm 2 has a Bernoulli reward with mean $3/4$. After cN steps, there is a constant probability, say p, that $\hat{\mu}_{1,cN} > \hat{\mu}_{2,cN}$. If this is the case, Algorithm 1 will start playing arm 1. Since the reward for arm 1 is deterministic, the estimate $\hat{\mu}_{1,t}$ does not change even on making more observations, and $\hat{\mu}_{2,t}$ does not change until it is played again (the algorithm does not make any new observations for this arm until it is played again). Therefore, the algorithm keeps playing arm 1 forever incurring a regret of at least $pT/2$.

Note that the drawback of this algorithm is that it does not explore enough, and stops exploring after a fixed time. Recall that any algorithm which is optimal for all time T needs to explore forever. The next algorithm tries to overcome that flaw.

3.3 ϵ-greedy Algorithm

One way to overcome this fixed period of exploration is to force our algorithm to always explore. More precisely, at every time step t, Algorithm 2 explores a random arm with some probability ϵ.

Algorithm 2 ϵ-greedy Algorithm

1: For all $t = 1, \ldots, N$
   (a) With probability $1 - \epsilon$, pull arm with highest estimate, i.e. $I_t = \text{argmax}_{i=1,\ldots,N} \ \hat{\mu}_{i,t}$
   (b) With probability ϵ, select a random arm with probability $1/N$ and pull it.

Unfortunately, Algorithm 2 also incurs a linear regret. Indeed, each arm i is pulled in average at least $\epsilon T/N$ times and therefore, the regret incurred is at least

$$\frac{\epsilon T}{N} \sum_{i:\mu^* > \mu_i} \Delta_i,$$

which is linear in T. Although Algorithm 2 only achieves linear regret, it does decently well in practice. Moreover, since it is very easy to implement, it is widely used. Furthermore, by smartly decreasing ϵ over time, one can actually achieve a polylogarithmic regret [2].

References

[1] Apostolos N. Burnetas, Michael N. Katehakis. Optimal Adaptive Policies for Sequential Allocation Problems, *Advances in Applied Mathematics*, 17(2):122–143, 1996.

[2] Nicolo Cesa-Bianchi and Paul Fischer. Finite-time regret bounds for the multiarmed bandit problem. *In Proceedings of ICML*, pages 100108, 1998

[3] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.