Algorithms and Uncertainty, Winter 2018/19 Lecture 15 (4 pages)

Stochastic Multi-Armed Bandits

Instructor: Thomas Kesselheim

So far, we have seen examples of online algorithms, in which we know nothing about the future; it can be arbitrary. We have also seen examples of Markov Decision processes, in which we have precise knowledge of the underlying stochastic process and its involved probabilities. Today, we will move to the space in between. There is an underlying stochastic process but we do not know it. In contrast to online algorithms, we can try to "learn" the involved probability distributions.

1 Model

Again we will have a version of a multi-armed bandit but there are no states. That is, whenever pulling arm i, the reward is an independent draw from some probability distribution. The crux is that we do not know these distributions. The only thing we can do is repeatedly pull arms, observe the reward, and make our decision which arms to pull based on the observed rewards up to this point.

In more detail, there are n arms with initially unknown reward probability distributions $\mathcal{D}_1, \dots, \mathcal{D}_n$ with means $\mu_1, \dots, \mu_n$. In step t, algorithm chooses an arm I_t and experiences reward $R_t \in [0,1]$ drawn from $\mathcal{D}_{I_t}$. So, the overall expected reward is $\mathbf{E}\left[\sum_{t=1}^{T} R_t\right]$.

If we had perfect knowledge of $\mathcal{D}_1, \dots, \mathcal{D}_n$, the expected reward would be maximized by always choosing arm i^* defined by $\mu_{i^*} = \max_i \mu_i$.[1] The *expected regret* of an algorithm is the difference between this expected reward and the one experienced by the algorithm

$$\text{Regret} = T \cdot \max_i \mu_i - \mathbf{E}\left[\sum_{t=1}^{T} R_t\right] \quad .$$

2 A Simple Explore-Exploit Algorithm

A very simple algorithm works as follows. There are two phases. In the *exploration* phase, we try out each arm exactly k times, so the length of this phase is kn steps. Afterwards, we have k samples from each distribution. From this, we can compute an empirical average $\hat{\mu}_i$ for each arm. If k is large enough, then $\hat{\mu}_i$ should be close to the actual μ_i. Therefore, in the *exploitation* phase (of length $T - kn$), we always play the arm that maximizes $\hat{\mu}_i$.

There is a clear trade-off between the exploration and the exploitation. If we set k too small, then $\hat{\mu}_i$ is computed only based on a few samples and can be far from μ_i. If k is too large, the exploration phase takes too long, during which we also play very bad arms.

Theorem 15.1. *Setting $k = T^{\frac{2}{3}}$, the expected regret of the simple algorithm is upper-bounded by $O(nT^{\frac{2}{3}} \ln nT)$.*

We leave the simple proof as an exercise.

[1] For simplicity we assume that this arm is unique.

3 UCB1 Algorithm

One of the major weaknesses of the simple algorithm is that it does not adapt the exploration to its observations: If an arm turns out to be very bad, intuitively it should be clear that it can be ignored after only a few samples.

The UCB1 algorithm is smarter. UCB stands for *upper confidence bound*. This almost explains the entire algorithm. We make no further distinction between exploration and exploitation. In each step, we use the empirical averages $\hat{\mu}_i^{(t)}$ of the arms observed so far. The actual means μ_i are close. They are closer, the more often we pulled the respective arm. Out of the empirical average and the number of times we pulled the arm so far, Q_i, we can compute confidence bounds, in which we expect the μ_i values to lie. We are optimistic and choose the arm with the highest upper bound on μ_i.

In more detail, in the first n steps, we pull each arm once. In any step $t > n$, we let $\hat{\mu}_i^{(t)}$ be the empirical average of arm i seen in steps *before* t. Furthermore, we let $Q_i^{(t)}$ denote the number of times that we pulled arm *before* step t.

From this, compute $\text{Index}_i^{(t)} = \hat{\mu}_i^{(t)} + \sqrt{\frac{\ln T}{Q_i^{(t)}}}$ and play the arm i with highest $\text{Index}_i^{(t)}$.

4 Distribution-Dependent Regret Bound

We will now derive a first bound on the expected regret of the UCB1 policy. Note that, with a more careful analysis, the constants could be improved.

Theorem 15.2. *The expected regret of UCB1 is at most* $\sum_{i \neq i^*} \frac{4 \ln T}{\Delta_i} + 4\Delta_i$*, where* $\Delta_i = \mu_{i^*} - \mu_i$*.*

Below, we will prove the following lemma.

Lemma 15.3. *For any arm* i*, let* $Q_i = Q_i^{(T+1)}$ *denote the number of times that arm* i *is pulled overall. Then* $\mathbf{E}[Q_i] \leq s_i + 4$*, where* $s_i = \frac{4 \ln T}{\Delta_i^2}$*.*

This lemma is actually all the work we have to do; the rest follows in a pretty straightforward way.

Proof of Theorem 15.2. Let $X_{i,t} = 1$ if the algorithm chooses to play i in step t. Furthermore, we define a potential reward $Y_{i,t}$, which the algorithm would get from pulling arm i in step t by drawing from each distribution $\mathcal{D}_1, \ldots, \mathcal{D}_n$. Importantly, these two random variables are independent. Therefore $\mathbf{E}[X_{i,t} Y_{i,t}] = \mathbf{E}[X_{i,t}] \mathbf{E}[Y_{i,t}] = \mathbf{E}[X_{i,t}] \mu_i$.

As $\mathbf{E}[R_t] = \mathbf{E}[\sum_{i=1}^n X_{i,t} Y_{i,t}]$, by linearity of expectation, the algorithm's expected reward is

$$\mathbf{E}\left[\sum_{t=1}^T R_t\right] = \mathbf{E}\left[\sum_{i=1}^n \sum_{t=1}^T X_{i,t} Y_{i,t}\right] = \sum_{i=1}^n \sum_{t=1}^T \mathbf{E}[X_{i,t}] \mu_i = \sum_{i=1}^n \mathbf{E}\left[\sum_{t=1}^T X_{i,t}\right] \mu_i = \sum_{i=1}^n \mathbf{E}[Q_i] \mu_i \ .$$

So, as always $\sum_{i=1}^n Q_i = T$, the expected regret is

$$T\mu_{i^*} - \mathbf{E}\left[\sum_{t=1}^T R_t\right] = \sum_{i=1}^n \mathbf{E}[Q_i](\mu_{i^*} - \mu_i) = \sum_{i=1}^n \mathbf{E}[Q_i] \Delta_i \ .$$

Using Lemma 15.3 and $\Delta_{i^*} = 0$, this immediately gives us the desired regret bound. □

5 Proof of Lemma 15.3

So, it only remains to prove Lemma 15.3. Interestingly, this only has to be a statement about one single arm i. We will have to show that the larger Δ_i the earlier the algorithm stops pulling the arm.

A major part of the work we do by proving the following lemma. It finally quantifies the frequently mentioned intuition that $\hat{\mu}_i^{(t)}$ is close to μ_i if arm i has been pulled often.

Lemma 15.4. *For all i, we have*

$$\mathbf{Pr}\left[\exists t : |\hat{\mu}_i^{(t)} - \mu_i| \geq \sqrt{\frac{\ln T}{Q_i^{(t)}}}\right] \leq \frac{2}{T} \ .$$

Proof. We will show this claim by using Hoeffding's inequality.

Lemma 15.5 (Hoeffding's inequality)**.** *Let $Z_1, \ldots, Z_N$ be independent random variables such that $a_i \leq Z_i \leq b_i$ with probability 1. Let $\bar{Z} = \frac{1}{N}\sum_{i=1}^N Z_i$ be their average. Then for all $\gamma \geq 0$*

$$\mathbf{Pr}\left[|\bar{Z} - \mathbf{E}\left[\bar{Z}\right]| \geq \gamma\right] \leq 2\exp\left(-\frac{2N^2\gamma^2}{\sum_{i=1}^N (b_i - a_i)^2}\right) \ .$$

The difficulty is that we cannot apply Hoeffding's inequality on a fixed $\hat{\mu}_i^{(t)}$: The value of $\hat{\mu}_i^{(t)}$ depends on how often we have pulled arm i so far and how the rewards turned out to be. If the first rewards were low, the value of $\hat{\mu}_i^{(t')}$ at the earlier time t' is low as well and arm i gets ignored.

Therefore, instead, we will consider the following way to determine all values of $\hat{\mu}_{i'}^{(t)}$. First, we draw, for each arm i', T values, which we denote by $X_{i',1}, \ldots, X_{i',T}$. Then, we determine $\hat{\mu}_{i'}^{(t)}$ from these draws in a deterministic way by following what the algorithm does; the j^{th} draw from arm i' has reward $X_{i',j}$.

Now, in order for there to be a t such that $|\hat{\mu}_i^{(t)} - \mu_i| \geq \sqrt{\frac{\ln T}{Q_i^{(t)}}}$, there has to be a k such that $|\frac{1}{k}\sum_{j=1}^k X_{i,j} - \mu_i| \geq \sqrt{\frac{\ln T}{k}}$. On these random variables $X_{i,1}, \ldots, X_{i,k}$, we can apply Hoeffding's inequality and get

$$\mathbf{Pr}\left[\left|\frac{1}{k}\sum_{j=1}^k X_{i,j} - \mu_i\right| \geq \sqrt{\frac{\ln T}{k}}\right] \leq 2\exp\left(-2k\frac{\ln T}{k}\right) = \frac{2}{T^2} \ .$$

Recall the union bound.

Lemma 15.6 (Union Bound)**.** *For any sequence of not necessarily disjoint events $\mathcal{E}_1, \mathcal{E}_2 \ldots$, we have*

$$\mathbf{Pr}\left[\mathcal{E}_1 \cup \mathcal{E}_2 \cup \ldots\right] \leq \mathbf{Pr}\left[\mathcal{E}_1\right] + \mathbf{Pr}\left[\mathcal{E}_2\right] + \ldots \ .$$

The event that there is a k for which $\left|\frac{1}{k}\sum_{j=1}^k X_{i,j} - \mu_i\right| \geq \sqrt{\frac{\ln T}{k}}$ is indeed such a union of events. Therefore

$$\mathbf{Pr}\left[\exists k : \left|\frac{1}{k}\sum_{j=1}^k X_{i,j} - \mu_i\right| \geq \sqrt{\frac{\ln T}{k}}\right] \leq \sum_{k=1}^T \mathbf{Pr}\left[\left|\frac{1}{k}\sum_{j=1}^k X_{i,j} - \mu_i\right| \geq \sqrt{\frac{\ln T}{k}}\right] \leq T\frac{2}{T^2} = \frac{2}{T} \ .$$

□

By plugging in the definition of $\text{Index}_i^{(t)}$, we immediately get that only with small probability an index $\text{Index}_i^{(t)}$ falls below μ_i or is much higher than this.

Corollary 15.7. *For all i, we have*

$$\mathbf{Pr}\left[\exists t : \text{Index}_i^{(t)} \leq \mu_i\right] \leq \frac{2}{T} \qquad \textit{and} \qquad \mathbf{Pr}\left[\exists t : \text{Index}_i^{(t)} \geq \mu_i + 2\sqrt{\frac{\ln T}{Q_i^{(t)}}}\right] \leq \frac{2}{T} \ .$$

This corollary is the key insight to compete our proof of Lemma 15.3.

Proof of Lemma 15.3. We can upper-bound $\mathbf{E}[Q_i]$ as follows.

$$\mathbf{E}[Q_i] \leq \mathbf{Pr}[Q_i \leq s_i]\, s_i + \mathbf{Pr}[Q_i > s_i]\, T \leq s_i + \mathbf{Pr}[Q_i > s_i]\, T \ .$$

So, if we are able to show that $\mathbf{Pr}[Q_i > s_i] \leq \frac{4}{T}$, we are done. Let us understand what has to happen so that $Q_i > s_i$ occurs. If we have $Q_i > s_i$, then there has to be some t_0 for which $Q_i^{(t_0)} = s_i$ and i is selected again. The algorithm selects i again because $\text{Index}_i^{(t_0)}$ is the highest index. So in particular $\text{Index}_i^{(t_0)} \geq \text{Index}_{i^*}^{(t_0)}$.

Let us now apply Corollary 15.7 to i and i^*. We get that

$$\mathbf{Pr}\left[\exists t : \text{Index}_{i^*}^{(t)} \leq \mu_{i^*}\right] \leq \frac{2}{T} \qquad \text{and} \qquad \mathbf{Pr}\left[\exists t : \text{Index}_i^{(t)} \geq \mu_i + 2\sqrt{\frac{\ln T}{Q_i^{(t)}}}\right] \leq \frac{2}{T} \ .$$

By union bound with probability at most $\frac{4}{T}$ one of the two events happen. If neither happens, then in particular at time t_0

$$\text{Index}_{i^*}^{(t_0)} > \mu_{i^*} \qquad \text{and} \qquad \text{Index}_i^{(t_0)} < \mu_i + 2\sqrt{\frac{\ln T}{s_i}} = \mu_i + \Delta_i = \mu_{i^*} \ .$$

This is a contradiction to $\text{Index}_i^{(t_0)} \geq \text{Index}_{i^*}^{(t_0)}$. This shows that $\mathbf{Pr}[Q_i > s_i] \leq \frac{4}{T}$. □

6 Distribution-Independent Regret Bound

The regret bound in Theorem 15.2 depends on the distributions. If there are arms with very similar means, the bound gets very poor. Fortunately, we can also derive a distribution-independent bound using the same techniques.

Theorem 15.8. *The expected regret of UCB1 is at most $5\sqrt{nT \ln T} + 4n$.*

Proof. We partition the arms as follows. Fix any $\epsilon > 0$ and let $S_1 = \{i \mid \Delta_i \leq \epsilon\}$, $S_2 = \{i \mid \Delta_i > \epsilon\}$. We can upper-bound the regret by

$$\epsilon T + \sum_{i \in S_2} \left(\frac{4 \ln T}{\Delta_i} + 4\Delta_i\right)$$

because whenever we pull an arm from S_1 the arm is at most ϵ worse and for S_2, we invoke the proof of Theorem 15.2. Note that $\Delta_i \leq 1$ because all rewards are in $[0, 1]$. Setting $\epsilon = \sqrt{\frac{n \ln T}{T}}$, we get

$$\epsilon T + \sum_{i \in S_2} \frac{4 \ln T}{\Delta_i} + 4\Delta_i \leq \sqrt{nT \ln T} + n\left((4 \ln T)\sqrt{\frac{T}{n \ln T}} + 4\right) = 5\sqrt{nT \ln T} + 4n \ . \qquad \square$$

Indeed, one can show that there are probability distributions for which the expected regret of any algorithm has to grow by at least $\Omega(\sqrt{T})$, so this guarantee is close to optimal.