CMSC 858G: Bandits, Experts and Games 09/12/16

Lecture 2: Bandits with i.i.d rewards (Part II)

Instructor: Alex Slivkins Scribed by: Amr Sharaf, Liqian Zhang

So far we've discussed non-adaptive exploration strategies. Now let's talk about adaptive exploration, in a sense that the bandit feedback of different arms in previous rounds are fully utilized.

Let's start with 2 arms. One fairly natural idea is to alternate them until we find that one arm is much better than the other, at which time we abandon the inferior one. But how to define "one arm is much better" exactly?

1 Clean event and confidence bounds

To flesh out the idea mentioned above, and to set up the stage for some other algorithms in this class, let's do some Probability with our old friend Hoeffding Inequality (HI).

Let $n_t(a)$ be the number of samples from arm a in round $1, 2, ..., t$; $\bar{\mu}_t(a)$ be the average reward of arm a so far. We would like to use HI to derive

$$\Pr\left(|\bar{\mu}_t(a) - \mu(a)| \leq r_t(a)\right) \geq 1 - \frac{2}{T^4}, \tag{1}$$

where $r_t(a) = \sqrt{\frac{2\log T}{n_t(a)}}$ is the confidence radius, and T is the time horizon.

In the intended application of HI, we have $n_t(a)$ independent random variables — one per each sample of arm a. Since HI requires a fixed number of random variables, (1) would follow immediately if $n_t(a)$ were fixed in advance. However, $n_t(a)$ is itself is a random variable. So we need a slightly more careful argument, presented below.

Let us imagine there is a tape of length T for each arm a, with each cell independently sampled from $\mathcal{D}_a$, as shown in Figure 1.

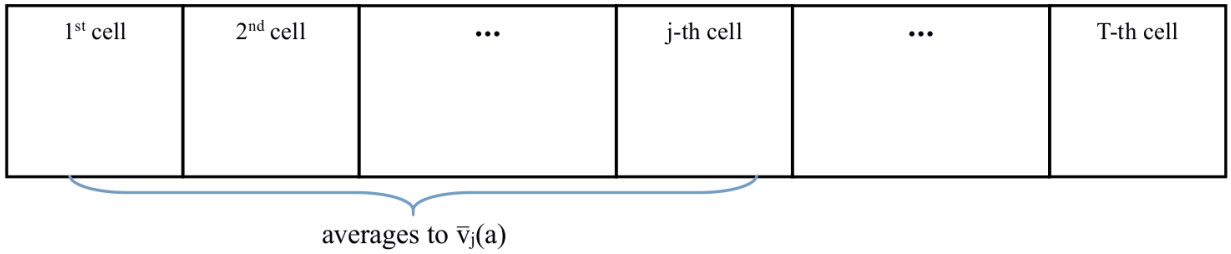

Figure 1: the j-th cell contains the reward of the j-th time we pull arm a, *i.e.,* reward of arm a when $n_t(a) = j$

Now one can use HI to derive that

$$\forall a \forall j \;\; \Pr\left(|\bar{v}_j(a) - \mu(a)| \leq r_t(a)\right) \geq 1 - \frac{2}{T^4}$$

where $\bar{v}_j(a)$ represents the average reward at arm a from first j times arm a is chosen. Taking a union bound, it follows that (assuming $K = \#\text{arms} \leq T$)

$$\Pr\left(\forall a \forall j \quad |\bar{v}_j(a) - \mu(a)| \leq r_t(a)\right) \geq 1 - \frac{2}{T^2}.$$

Now, observe that this event is precisely the same as

$$\mathcal{E} := \left\{\forall a \forall t \quad |\bar{\mu}_t(a) - \mu(a)| \leq r_t(a)\right\}. \tag{2}$$

Therefore, we have proved:

Lemma 1.1. $\Pr[\mathcal{E}] \geq 1 - \frac{2}{T^2}$, *where* $\mathcal{E}$ *is given by* (2).

The event in (2) will be called the *clean event*.

Under the assumption of clean event, we can now draw the range of average reward for arm a at round t as in Figure 2. We introduce the concepts of *upper confidence bound* $\text{UCB}_t(a) = \bar{\mu}_t(a) + r_t(a)$, and *lower confidence bound* $\text{LCB}_t(a) = \bar{\mu}_t(a) - r_t(a)$. These concepts will be used in several algorithms in this course.

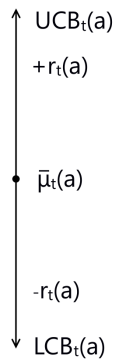

Figure 2: The true expected reward $\mu(a)$ will fall in the interval centered at $\bar{\mu}_t(a)$ with radius $r_t(a)$ under clean event

2 Successive Elimination

Let's recap our idea for two arms: alternate them until we find that one arm is much better than the other. Now, using the confidence bounds, we can naturally define the criterion of "one arm is much better". The full algorithm for two arms is as follows:

we alternate two arms until $\text{UCB}_t(a) < \text{LCB}_t(a')$ (check the condition after the even rounds). When the condition is met, we abandon a, use a' forever.

For analysis, assume the clean event. Note that the "disqualified" arm cannot be the best arm. But how much regret do we accumulate *before* disqualifying one arm?

Let t be the last round when we did *not* invoke the stopping rule, i.e., when the confidence intervals of the two arms still overlap (see Figure 3). Then

$$\Delta := |\mu(a) - \mu(a')| \leq 2(r_t(a) + r_t(a')).$$

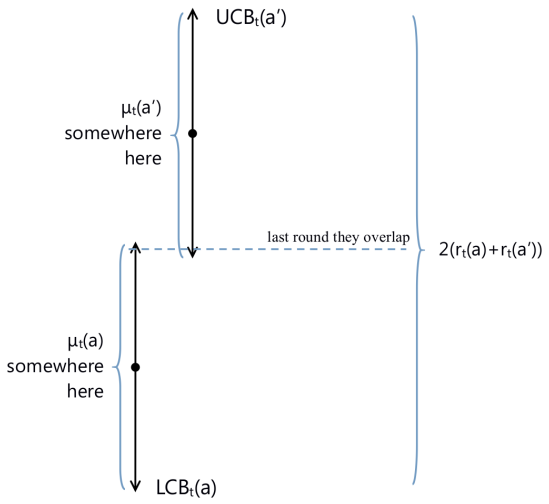

Figure 3: t is the last round that the two confidence intervals still overlap

Since we've been alternating the two arms before time t, we have $n_t(a) = \frac{t}{2}$ (up to floor and ceiling), which yields

$$\Delta \leq 2(r_t(a) + r_t(a')) \leq 4\sqrt{\frac{2\log T}{\lfloor t/2 \rfloor}} = O\left(\sqrt{\frac{\log T}{t}}\right).$$

Then total regret accumulated till round t is

$$R(t) \leq \Delta \times t \leq O(t \cdot \sqrt{\frac{\log T}{t}}) = O(\sqrt{t \log T})).$$

Since we've chosen the best arm from then on, we have $R(T) \leq O(\sqrt{T \log T})$.

To complete the proof, we need to argue that the "bad event" $\bar{\mathcal{E}}$ contributes a negligible amount to regret (much like we did for explore-first):

$$\begin{aligned}
\mathbb{E}[R(T)] &= \mathbb{E}[R(T)|\text{clean event}] \times \Pr[\text{clean event}] \ + \ \mathbb{E}[R(T)|\text{bad event}] \times \Pr[\text{bad event}] \\
&\leq \mathbb{E}[R(T)|\text{clean event}] + T \times O(T^{-2}) \\
&\leq O(\sqrt{T \log T}).
\end{aligned}$$

This completes the analysis for $K = 2$ arms.

□

This stopping rule can also be generalized to K arms.

```
1 Initially all arms are set "active";
2 Each phase:
3     try all active arms (thus each phase may contain multiple rounds);
4     deactivate all arms a s.t. ∃arm a' with UCB_t(a) < LCB_t(a');
5 Repeat until end of rounds.
```

Algorithm 1: Successive Elimination

To analyze the performance of this algorithm, it suffices to focus on the clean event (2) (indeed, as in the case of $k = 2$ arms, the contribution of the "bad event" $\bar{\mathcal{E}}$ can be neglected).

Let a^* be an optimal arm, and consider any arm a such that $\mu(a) < \mu(a^*)$. Look at the last round t when we did not deactivate arm a yet (or the last round T if a is still active at the end). As in the argument for two arms, the confidence intervals of the two arms a and a^* before round t must overlap. Therefore:

$$\Delta(a) := \mu(a^*) - \mu(a) \leq 2(r_t(a^*) + r_t(a)) = O(r_t(a)).$$

The last equality is because $n_t(a)$ and $n_t(a^*)$ differ at most 1, as the algorithm has been alternating active arms. Since arm a is never played after round t, we have $n_t(a) = n_T(a)$, and therefore $r_t(a) = r_T(a)$.

We have proved the following crucial property:

$$\Delta(a) \leq O(r_T(a)) = O\left(\sqrt{\frac{\log T}{n_T(a)}}\right) \quad \text{for each arm } a \text{ with } \mu(a) < \mu(a^*). \tag{3}$$

The rest of the analysis will only rely on property (3). In other words, it does not matter which algorithm has achieved this property.

The contribution of arm a to regret, denoted $R(T;a)$, can be expressed as $\Delta(a)$ for each round this arm is played; by (3) we can bound this quantity as

$$R(T;a) \leq n_T(a) \cdot O\left(\sqrt{\frac{\log T}{n_T(a)}}\right) = O(\sqrt{n_T(a)\log T}).$$

Let $\mathcal{A}$ be the set of all K arms, and let $\mathcal{A}^+ = \{a : \mu(a) < \mu(a^*)\}$ be the set of all arms that contribute to regret. Summing up over all arms $a \in \mathcal{A}^+$, we obtain:

$$R(T) = \sum_{a\in\mathcal{A}^+} R(T;a) = O(\sqrt{\log T}) \sum_{a\in\mathcal{A}^+} \sqrt{n_T(a)} \leq O(\sqrt{\log T}) \sum_{a\in\mathcal{A}} \sqrt{n_T(a)}. \tag{4}$$

Note that $f(x) = \sqrt{x}$ is a real continuous concave function, and $\sum_{a\in\mathcal{A}} n_T(a) = T$, by Jensen's Inequality we have

$$\frac{1}{K}\sum_{a\in\mathcal{A}} \sqrt{n_T(a)} \leq \sqrt{\frac{1}{K}\sum_{a\in\mathcal{A}} n_T(a)} = \sqrt{\frac{T}{K}}$$

Plugging this into (4), we see that $R(T) \leq O(\sqrt{KT\log T})$. Thus, we have proved:

Theorem 2.1. *Successive Elimination algorithm achieves regret $\mathbb{E}[R(T)] = O(\sqrt{KT\log T})$, where K is the number of arms and T is the time horizon.*

Remark 2.2. The $\sqrt{T}$ dependence on T in the regret bound for Successive Elimination should be contrasted with the $T^{2/3}$ dependence for Explore-First. This improvement is possible due to adaptive exploration.

We can also use (3) to obtain another regret bound. Rearranging the terms in (3), we obtain $n_T(a) \leq O\left(\frac{\log T}{[\Delta(a)]^2}\right)$. Therefore, for each arm $a \in \mathcal{A}^+$ we have:

$$R(T;a) = \Delta(a) \cdot n_T(a) \leq \Delta(a) \cdot O\left(\frac{\log T}{[\Delta(a)]^2}\right) = O\left(\frac{\log T}{\Delta(a)}\right). \tag{5}$$

Summing up over all arms $a \in \mathcal{A}^+$, we obtain:

$$R(T) \leq O(\log T) \left[\sum_{a \in \mathcal{A}^+} \frac{1}{\Delta(a)} \right].$$

Theorem 2.3. *Successive Elimination algorithm achieves regret*

$$\mathbb{E}[R(T)] \leq O(\log T) \left[\sum_{\text{arms } a \text{ with } \mu(a) < \mu(a^*)} \frac{1}{\mu(a^*) - \mu(a)} \right]. \tag{6}$$

Remark 2.4. Regret of Successive Elimination is (at most) logarithmic in T with an *instance-dependent constant*. The latter constant can be arbitrarily large, depending on a problem instance. The distinction between regret bounds achievable with an absolute constant (as in Theorem 2.1) and regret bounds achievable with an instance-dependent constant is typical for multi-armed bandit problems. The existence of the logarithmic regret bound is another benefit of adaptive exploration compared to non-adaptive exploration.

Remark 2.5. It is instructive to derive Theorem 2.1 in a different way: starting from the logarithmic regret bound in (5). Informally, we need to get rid of arbitrarily small $\Delta(a)$'s in the denominator. Let us fix some $\epsilon > 0$, then regret consists of two parts:

- all arms a with $\Delta(a) \leq \epsilon$ contribute at most ϵ per round, for a total of ϵT;
- each arms a with $\Delta(a) > \epsilon$ contributes at most $R(T; a) \leq O(\frac{1}{\epsilon} \log T)$ to regret; thus, all such arms contribute at most $O(\frac{K}{\epsilon} \log T)$.

Combining these two parts, we see that (assuming the clean event)

$$R(T) \leq O\left(\epsilon T + \frac{K}{\epsilon} \log T \right).$$

Since this holds for $\forall \epsilon > 0$, we can choose the ϵ that minimizes the right-hand side. Ensuring that $\epsilon T = \frac{K}{\epsilon} \log T$ yields $\epsilon = \sqrt{\frac{K}{T} \log T}$, and therefore $R(T) \leq O(\sqrt{KT \log T})$.

3 UCB1 Algorithm

Let us consider another approach for adaptive exploration. The algorithm is based on the *optimism under uncertainty* principle: assume each arm is as good as it can possibly be given the observations so far, and choose the best arm based on these optimistic estimates. This intuition leads to the following simple algorithm called UCB1:

1 Try each arm once;
2 In each round t, pick $\operatorname{argmax}_{a \in \mathcal{A}} \text{UCB}_t(a)$, where $\text{UCB}_t(a) = \bar{\mu}_t(a) + r_t(a)$;

Algorithm 2: UCB1 Algorithm

Remark 3.1. Here's more intuition for this algorithm. An arm a is chosen in round t because it has a large $\text{UCB}_t(a)$. The latter can happen for two reasons: (i) $\bar{\mu}_t(a)$ is large, implying a high reward; or (ii) $r_t(a)$ is large, *i.e.,* $n_t(a)$ is small, implying an under-explored arm. In both cases, this arm is worth choosing. Further, the $\bar{\mu}_t(a)$ and $r_t(a)$ summands represent exploitation and exploration, resp., and summing them up is a natural way to implement exploration-exploitation tradeoff.

To analyze this algorithm, let us focus on the clean event (2), as before. Let us use the same notation as before: a^* is a best arm, and a_t is the arm chosen by the algorithm in round t. According to the algorithm, $\text{UCB}_t(a_t) \geq \text{UCB}_t(a^*)$. Under clean events, $\mu(a_t) + r_t(a_t) \geq \bar{\mu}_t(a_t)$ and $\text{UCB}_t(a^*) \geq \mu(a^*)$. Therefore:

$$\mu(a_t) + 2r_t(a_t) \geq \bar{\mu}_t(a_t) + r_t(a_t) = \text{UCB}_t(a_t) \geq \text{UCB}_t(a^*) \geq \mu(a^*).$$

It follows that

$$\Delta(a_t) := \mu(a^*) - \mu(a_t) \leq 2r_t(a_t) = 2\sqrt{\frac{2\log T}{n_t(a_t)}}. \tag{7}$$

Remark 3.2. This is a very cute trick, which resurfaces in the analyses of several UCB-like algorithms for more general bandit settings.

In particular, for each arm a consider the last round t when this arm is chosen by the algorithm. Applying (7) to this round gives us property (3). The rest of the analysis follows from that property, as in the analysis of Successive Elimination.

Theorem 3.3. *Algorithm UCB1 achieves regret* $\mathbb{E}[R(T)] \leq O(\sqrt{TK\log T}$ *and also the logarithmic regret bound in* (6).

4 Some remarks

Bibliography. Successive Elimination is from Even-Dar et al. (2002), and UCB1 is from Auer et al. (2002).

Regret for all rounds at once. So far we considered a fixed time horizon T. What if it is not known in advance, and we wish to achieve good regret bounds for all rounds t? In all algorithms we have considered, the knowledge of the time horizon T is needed (only) in the definition of the confidence radius $r_t(a)$. There are several remedies:

- Suppose we wish to achieve same regret bounds for all rounds $t \leq T$, for some time horizon T that is *not* known to the algorithm. If an upper bound $T' \geq T$ is known, then one can use T' instead of T in the definition of the confidence radius. Since $r_t(a)$ depends on T only logarithmically, a rather significant over-estimates can be tolerated.

- In UCB1, one can use an "online" version of confidence radius: $r_t(a) = \sqrt{\frac{2\log t}{n_t(a)}}$. In fact, this is the original version of UCB1 from (Auer et al., 2002). It achieves the same regret bounds, and with better constants, at the cost of a somewhat more complicated analysis.

- Any algorithm for known time horizon can be converted to an algorithm for arbitrary time horizon using the *doubling trick*. Here, the new algorithm proceeds in phases of exponential duration. Each phase $i = 1, 2, \ldots$ lasts 2^i rounds, and executes a fresh run of the original algorithm. This approach achieves the "right" theoretical guarantees (more on this on the homework). However, it is somewhat impractical because the algorithm needs to "forget" everything it has learned after each phase.

Instantaneous regret. In this lecture we discussed cumulative regret. Another type of regret is *instantaneous regret* (also called *simple regret*), defined as $\mathbb{E}[r(t)] = \mu^* - \mathbb{E}[\mu(a_t)]$. In particular, it may be desirable to spread the regret "more uniformly" over rounds, so as to avoid spikes in instantaneous regret. Then (in addition to a "good" cumulative regret) one would also like an upper bound on instantaneous regret that decreases monotonically over time.

Bandits with predictions. The standard goal for bandit algorithms is (as discussed in this lecture) to maximize cumulative reward. An alternative goal is to output a prediction a_t^* after each round t. The algorithm is then graded only on the quality of these predictions (so that it does not matter how much reward is accumulated). There are two standard ways to make this objective formal: (i) minimize instantaneous regret $\mu^* - \mu(a_T^*)$, and (ii) maximize the probability of choosing the best arm: $\Pr[a_t^* = a^*]$. Essentially, good algorithms for cumulative regret, such as Successive Elimination and UCB1, are also good for this version (more on the homework). However, improvements are possible in some regimes, and there are some papers on that.

References

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC bounds for multi-armed bandit and Markov decision processes. In *15th Conf. on Learning Theory (COLT)*, pages 255–270, 2002.