CMSC 858G: Bandits, Experts and Games 10/10/16

Lecture 6: Lipschitz Bandits

Instructor: Alex Slivkins Scribes: Shuo Li, Sheng Yang

Motivation: similarity between arms. In various bandit problems, we may have information on similarity between arms, in the sense that 'similar' arms have similar expected rewards. For example, arms can correspond to "items" (e.g., documents) with feature vectors, and similarity can be expressed as some notion of distance between feature vectors. Another example would be the dynamic pricing problem, where we have numerical similarity between prices, and it is often assumed that similar prices correspond to similar expected rewards.

Throughout this lecture, we will consider bandits with IID rewards: the reward for each arm x is an independent sample from some fixed distribution with expectation $\mu(x)$. We use x, y to denote arms, as they will also denote "points" on a line or in a metric space.

1 Continuum-armed bandits (CAB)

Let us start with a special case called *continuum-armed bandits* (CAB). In this problem, the set of arms is $X = [0, 1]$, and we have a *Lipschitz Condition*:

$$|\mu(x) - \mu(y)| \leq L \cdot |x - y| \quad \text{for any two arms } x, y \in X, \tag{1}$$

where L is a constant known to the algorithm.[1]

Note that we have infinitely many arms, and in fact, *continuously* many. While the MAB problem with infinitely many arms is hopeless in general, CAB turns out tractable due to the Lipschitz condition.

1.1 Simple solution: fixed discretization

The idea is to pick a fixed, finite set of arms $S \subset X$, and use this set as an "approximation" for the full set X. Then we focus only on arms in S, and run an off-the-shelf MAB algorithm ALG, such as UCB1 or Successive Elimination, that only considers these arms. In this context, the set S is called a *discretization* of X.

The best arm in S will be denoted $\mu^*(S) = \sup_{x \in S} \mu(x)$. In each round, algorithm ALG can only hope to approach expected reward $\mu^*(S)$, and additionally suffers *discretization error*

$$\texttt{DE}(S) = \mu^*(X) - \mu^*(S). \tag{2}$$

More formally, we can represent regret of ALG as a sum

$$\begin{aligned} R(T) &= \mu^*(X) - W(ALG) \\ &= (\mu^*(S) - W(ALG)) + (\mu^*(X) - \mu^*(S)) \\ &= R_S(T) + \texttt{DE}(S), \end{aligned}$$

[1]A function $\mu : X \to \mathbb{R}$ which satisfies (1) is called *Lipschitz-continuous* on X, with *Lipschitz constant* L.

where $W(ALG)$ is the total reward of the algorithm, and $R_S(T)$ is the regret relative to $\mu^*(S)$.

Suppose ALG attains optimal regret $O(\sqrt{KT\log T})$ on any problem instance with time horizon T and K arms. Then

$$\mathbb{E}[R(T)] \leq O(\sqrt{|S|T\log T}) + \texttt{DE}(S)\cdot T. \tag{3}$$

Thus, we have a tradeoff between the size of S and its discretization error.

One simple and natural solution is to use *uniform discretization* with k arms: divide the interval $[0,1]$ into intervals of fixed length $\epsilon = \frac{1}{k-1}$. Then

$$S = \{i\cdot\epsilon : i = 0,1,2,\ \dots\ ,k-1\} \subset X.$$

It is easy to see that $\texttt{DE}(S) \leq L\epsilon$. Picking $\epsilon = (TL^2/\log T)^{-1/3}$, we obtain

$$\mathbb{E}[R(T)] \leq O\left(L^{1/3}\cdot T^{2/3}\cdot \log^{1/3}(T)\right).$$

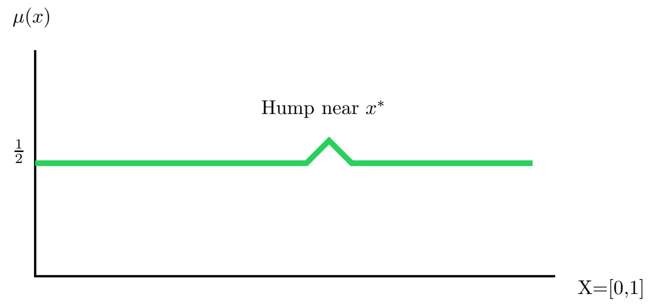
1.2 Lower Bound

The simple solution described above is in fact optimal in the worst case: we have an $\Omega(T^{2/3})$ lower bound on regret. We prove this lower bound via a relatively simple reduction from the "main lower bound": the $\Omega(\sqrt{KT})$ lower bound that we've seen in Lecture 4 (henceforth called `MainLB`).

The new lower bound will involve problem instances with 0-1 rewards such that all arms have mean reward $\mu(x) = \frac{1}{2}$ except those near the unique best arm x^*, whose mean reward is $\mu(x^*) = \frac{1}{2}+\epsilon$. More formally, let us define a problem instance $\mathcal{I}(x^*,\epsilon)$ by

$$\mu(x) = \begin{cases} \frac{1}{2}, & |x-x^*| \geq \epsilon/L \\ \frac{1}{2}+\epsilon - L\cdot|x-x^*|, & \text{otherwise.} \end{cases} \tag{4}$$

It is easy to see that any such problem instance satisfies the Lipschitz Condition (1). Note that we need to have a small "bump" near x^*, due to the Lipschitz Condition. Informally, we will refer to $\mu(\cdot)$ as the "bump function":



The lower bound for CAB is stated as follows:

Theorem 1.1 (Lower bound for CAB)**.** *Consider CAB with time horizon T. Let ALG be any algorithm for this problem. There exists a problem instance $\mathcal{I} = \mathcal{I}(x^*,\epsilon)$ such that*

$$\mathbb{E}\left[R(T) \mid \mathcal{I}\right] \geq \Omega(T^{2/3}). \tag{5}$$

The intuition is as follows. Fix $K \in \mathbb{N}$, let $\epsilon = 1/2K$, and assume $L = 1$ for simplicity. Focus on problem instances $\mathcal{I}(a^*/K, \epsilon)$, where $a^* \in [K] := \{1, \ldots, K\}$. Thus, the set of arms $X = [0,1]$ is partitioned into K disjoint intervals of length 2ϵ, and the 'bump' can be in any one of these intervals. Further, whenever an algorithm chooses an arm x in one such interval, choosing an arm in the *center* of the same interval appears to be the best option: either this interval contains a bump, and the center is the best arm, or all arms in this interval have the same mean reward $\frac{1}{2}$. But if an algorithm only chooses among arms that are multiples of $1/K$, we have K-armed bandit problem instances where all arms have mean reward $\frac{1}{2}$ except one with mean reward $\frac{1}{2} + \epsilon$. Which is precisely the family of instances from `MainLB`. Then one can apply the lower bound from `MainLB`, and tune the parameters to obtain (5). To turn this intuition into a proof, the main obstacle is to argue that "choosing the center of the same interval is the best option".

Let us recap `MainLB`, restating it in a way that is convenient for this lecture. Recall that `MainLB` considered problem instances $\mathcal{J}(a^*, \epsilon)$ with 0-1 rewards such that all arms a have mean reward $\frac{1}{2}$, except the best arm a^* whose mean reward is $\frac{1}{2} + \epsilon$.

Theorem 1.2 (`MainLB`). *Consider bandits with IID rewards, with K arms and time horizon T (for any K, T). Let ALG be any algorithm for this problem. Pick any positive $\epsilon \le \sqrt{cK/T}$, where c is an absolute constant from Lemma 1.1 in Lecture 4. Then there exists a problem instance $\mathcal{J} = \mathcal{J}(a^*, \epsilon)$*

$$\mathbb{E}\left[R(T) \mid \mathcal{J}\right] \ge \Omega(\epsilon T).$$

To prove Theorem 1.1, we show how to reduce the problem instances from `MainLB` to CAB in a way that we can apply `MainLB` and derive the claimed lower bound (5). On a high level, our plan is as follows: (i) take any problem instance $\mathcal{J}$ from `MainLB` and "embed" it into CAB, and (ii) show that any algorithm for CAB will, in fact, need to solve $\mathcal{J}$, (iii) tune the parameters to derive (5).

Proof of Theorem 1.1. For simplicity of exposition, assume the Lipschitz constant is $L = 1$ (arbitrary L is treated similarly). Let $K \in \mathbb{N}$ be a parameter to be fixed later, and let $\epsilon = 1/2K$.

We will consider problem instances from `MainLB` with K arms and the same time horizon T; the set of arms will be denoted as $[K] = \{1, 2, \ldots, K\}$. Each problem instance $\mathcal{J} = \mathcal{J}(a^*, \epsilon)$ from `MainLB` will correspond to an instance $\mathcal{I} = \mathcal{I}(x^*, \epsilon)$ of CAB with $x^* = a^*/K$; in particular, each arm $a \in [K]$ in $\mathcal{J}$ corresponds to an arm $x = a/K$ in $\mathcal{I}$. We view $\mathcal{J}$ as a discrete version of $\mathcal{I}$; in particular, note that $\mu_{\mathcal{J}}(a) = \mu(a/K)$, where $\mu(\cdot)$ is the reward function for $\mathcal{I}$, and $\mu_{\mathcal{J}}(\cdot)$ is the reward function for $\mathcal{J}$.

Let us consider ALG as applied to problem instance $\mathcal{I}$ of CAB, and use it as a subroutine to construct an algorithm ALG' which solves instance $\mathcal{J}$ of K-armed bandits. Each round in algorithm ALG' proceeds as in the following table:

ALG for CAB instance $\mathcal{I}$	ALG' for K-armed bandits instance $\mathcal{J}$
chooses arm $x \in \left[\frac{a}{K} - \epsilon,\ \frac{a}{K} + \epsilon\right)$, $a \in [K]$	
	chooses arm a
	receives reward $r \in \{0,1\}$ with mean $\mu_{\mathcal{J}}(a)$
receives reward $r_x \in \{0,1\}$ with mean $\mu(x)$	

In other words, ALG is called and returns some arm $x \in [0,1]$. This arm falls into the interval $\left[\frac{a}{K} - \epsilon,\ \frac{a}{K} + \epsilon\right)$, for some $a \in [K]$. Then algorithm ALG' chooses arm a. When ALG' receives reward r, it uses r and x to compute reward $r_x \in \{0,1\}$ such that $\mathbb{E}[r_x \mid x] = \mu(x)$, and feed it back to ALG.

To complete the construction, it remains to define r_x so that $\mathbb{E}[r_x \mid x] = \mu(x)$. and r_x can be computed using information available to ALG' in a given round. (In particular, the computation of r_x cannot use $\mu_{\mathcal{J}}(a)$ or $\mu(x)$, since they are not know to ALG'.) We define r_x as follows:

$$r_x = \begin{cases} r & \text{with probability } p_x \\ X & \text{otherwise} \end{cases} \quad \text{where} \quad p_x = 1 - \left| x - \tfrac{a}{K} \right| / \epsilon, \tag{6}$$

and X is an independent toss of an unbiased random coin, i.e., a 0-1 random variable with expectation $\frac{1}{2}$. Then

$$\begin{aligned} \mathbb{E}[r_x \mid x] &= p_x \cdot \mu_{\mathcal{J}}(a) + (1-p_x) \cdot \tfrac{1}{2} \\ &= \tfrac{1}{2} + \left(\mu_{\mathcal{J}}(a) - \tfrac{1}{2}\right) \cdot p_x \\ &= \begin{cases} \tfrac{1}{2} & \text{if } a \neq a^* \\ \tfrac{1}{2} + \epsilon p_x & \text{if } a = a^* \end{cases} \\ &= \mu(x). \end{aligned}$$

For each round t, let x_t and a_t be the arms chosen by ALG' and ALG, respectively. By construction, the mean reward of x_t cannot exceed that of a_t; rigorously:

$$\mathbb{E}[\mu_{\mathcal{J}}(a_t) \mid x_t] \geq \mathbb{E}[\mu(x_t)].$$

It follows that the total expected reward of ALG on instance $\mathcal{I}$ cannot exceed that of ALG' on instance $\mathcal{J}$. Since the best arms in both problem instances have the same expected rewards $\frac{1}{2} + \epsilon$, it follows that

$$\mathbb{E}[R(T) \mid \mathcal{I}] \geq \mathbb{E}[R'(T) \mid \mathcal{J}],$$

where $R(T)$ and $R'(T)$ denote regret of ALG and ALG', respectively.

Recall that algorithm ALG' can solve any K-armed bandits instance of the form $\mathcal{J} = \mathcal{J}(a^*, \epsilon)$. Let us apply Theorem 1.2 to derive a lower bound on the regret of ALG'. Specifically, let us fix $K = (T/4c)^{1/3}$, so as to ensure that $\epsilon \leq \sqrt{cK/T}$, as required in Theorem 1.2. Then there exists some instance $\mathcal{J} = \mathcal{J}(a^*, \epsilon)$ such that

$$\mathbb{E}[R'(T) \mid \mathcal{J}] \geq \Omega(\sqrt{\epsilon T}) = \Omega(T^{2/3})$$

Thus, taking the corresponding instance $\mathcal{I}$ of CAB, we conclude that $\mathbb{E}[R(T) \mid \mathcal{I}] \geq \Omega(T^{2/3})$. □

2 Lipschitz MAB

The *Lipschitz MAB Problem* is a multi-armed bandit problem with IID rewards, where the reward function μ satisfies the *Lipschitz condition*:

$$|\mu(x) - \mu(y)| \leq \mathcal{D}(x,y) \quad \text{for any two arms } x, y. \tag{7}$$

Here $\mathcal{D}$ is a metric on arms which is known to the algorithm. The set of arms X can be finite or infinite. Recall that $\mu(x)$ denotes the mean reward of arm x, and the realized rewards are in $[0,1]$.

Informally, the metric $\mathcal{D}$ defines a notion of similarity between arms such that similar arms have similar mean rewards. Note that w.l.o.g. $\mathcal{D}(x,y) \leq 1$.

2.1 (Very brief) background on metric spaces

A Metric Space is a pair $(X, \mathcal{D})$, where $\mathcal{D}$ is a *metric*: a function $\mathcal{D} : X \times X \to \mathbb{R}$ which represents "distance" between the elements of X. Formally, a metric satisfies the following axioms:

- Non-negativity: $\mathcal{D}(x, y) \geq 0$.
- Zero iff equal: $\mathcal{D}(x, y) = 0$ iff $x = y$
- Symmetry: $\mathcal{D}(x, y) = \mathcal{D}(y, x)$
- Triangle inequality: $\mathcal{D}(x, y) + \mathcal{D}(y, z) \geq \mathcal{D}(x, z)$

For $Y \subset X$, the pair $(Y, \mathcal{D})$ is also a metric space, where (by a slight abuse of notation) $\mathcal{D}$ denotes the same metric restricted to the points in Y.

Some examples:

- in CAB, $X = [0, 1]$ and the metric is $\mathcal{D}(x, y) = |x - y|$.
- more generally, one could consider $X = [0, 1]^d$, and the distance function is the p–norm, $p \geq 1$:

  $$\ell_p(x, y) = \|x - y\|_p := \left(\sum_{i=1}^{d} (x_i - y_i)^p \right)^{1/p}.$$

  Most commonly considered are ℓ_1-metric (a.k.a. Manhattan metric) and ℓ_2-metric (a.k.a. Euclidean distance).
- X can be any subset of $[0, 1]^d$, finite or infinite, under the same distance function ℓ_p, $p \geq 1$.
- The shortest-path metric of a graph: X is the nodes of a graph, and $\mathcal{D}(x, y)$ is the length of a shortest path between x and y.

2.2 Fixed discretization

The developments in Section 1.1 carry over word-by-word, up until (3). As before, for a given $\epsilon > 0$ we would like to pick a subset $S \subset X$ with discretization error $\mathtt{DE}(S) \leq \epsilon$ and $|S|$ as small as possible.

Example 2.1. Suppose the metric space is $X = [0, 1]^d$ under the ℓ_p metric, $p \geq 1$. Let us extend the notion of *uniform discretization* from Section 1.1: consider a subset $S \subset X$ that consists of all points whose coordinates are multiples of a given $\epsilon > 0$. Then S consists of $(\lceil 1/\epsilon \rceil)^d$ points, and its discretization error is $\mathtt{DE}(S) \leq c_{p,d} \cdot \epsilon$, where $c_{p,d}$ is a constant that depends only on p and d (e.g., $c_{p,d} = d$ for $p = 1$). Plugging this into (3), we obtain

$$\mathbb{E}[R(T)] \leq O\left(\sqrt{(\tfrac{1}{\epsilon})^d \cdot T \log T} + \epsilon T \right) = O\left(T^{\frac{d+1}{d+2}} \, (c \log T)^{\frac{1}{d+2}} \right), \tag{8}$$

where the last equality holds if we take $\epsilon = (\frac{\log T}{T})^{1/(d+2)}$.

As it turns out, the $\tilde{O}(T^{(d+1)/(d+2)})$ regret in the above example is typical for Lipschitz MAB problem. However, we will need to define a suitable notion of "dimension" that will apply to an arbitrary metric space.

Let us generalize the notion of uniform discretization to an arbitrary metric space.

Definition 2.2. A subset $S \subset X$ is called an *ϵ-mesh* if $\forall x \in X\ \exists y \in S$ such that $\mathcal{D}(x, y) \leq \epsilon$.

In words, S is an ϵ-mesh if every point in X is within distance ϵ from some point in S. It is easy to see that $\mathtt{DE}(S) \leq \epsilon$. In what follows, we characterize the smallest size of an ϵ-mesh, using some notions from the analysis of metric spaces.

Definition 2.3 (ϵ-covering and covering number)**.** A collection of subsets $X_i \subset X$ such that the diameter of each subset is at most ϵ and $X = \bigcup X_i$. The smallest number of subsets in an ϵ-covering is called the *covering number* and denoted $N_\epsilon(X)$.

The *diameter* of a metric space is the maximal distance between any two points; the diameter of a subset $X' \subset X$ is the diameter of $(X', \mathcal{D})$, as long as the metric $\mathcal{D}$ is clear from the context.

The covering number characterizes the "complexity" of a metric space (in a specific sense that happens to be relevant to Lipschitz MAB). This notion of complexity is "robust", in the sense that a subset $X' \subset X$ cannot be more "complicated" than X: indeed, $N_\epsilon(X') \leq N_\epsilon(X)$.

Fact 2.4. *Consider an ϵ-covering $\{X_1, \ldots, X_N\}$. For each subset X_i, pick an arbitrary representative $x_i \in X_i$. Then $S = \{x_1, \ldots, x_N\}$ is an ϵ-mesh.*

Thus, for each $\epsilon > 0$ there exists an ϵ-mesh of size $N_\epsilon(X)$, so we can plug it into (3). But how do we optimize over all $\epsilon > 0$? It is convenient to characterize $N_\epsilon(X)$ for all ϵ simultaneously:

Definition 2.5. The *covering dimension* of X, with multiplier $c > 0$, is

$$\mathtt{COV}_c(X) = \inf_{d \geq 0} \left\{ N_\epsilon(X) \leq \frac{c}{\epsilon^d} \quad \forall \epsilon \geq 0 \right\}.$$

In particular, the covering dimension in Example 2.1 is d (with an appropriate multiplier). The covering dimension of a subset $X' \subset X$ cannot exceed that of X: indeed, $\mathtt{COV}_c(X') \leq \mathtt{COV}_c(X)$.

Now one can plug in $|S| = N_\epsilon(X) \leq c/\epsilon^d$ into (3), and do the same computation as in Example 2.1. Therefore, we obtain the following theorem:

Theorem 2.6 (fixed discretization)**.** *Consider Lipschitz MAB problem on a metric space $(X, \mathcal{D})$, with time horizon T. Fix any $c > 0$, and let $d = \mathtt{COV}_c(X)$ be the covering dimension. There exists a subset $S \subset X$ such that if one runs UCB1 using S as a set of arms, one obtains regret*

$$\mathbb{E}[R(T)] = O\left(T^{\frac{d+1}{d+2}} \, (c \log T)^{\frac{1}{d+2}} \right).$$

There is a matching lower bound of $\Omega\left(T^{(d+1)/(d+2)}\right)$. This lower bound can be achieved on $X = [0,1]^d$, with any ℓ_p-metric, $p \geq 1$. It can be proved similarly to Theorem 1.1.

3 Adaptive discretization: the Zooming Algorithm

Despite the existence of a matching lower bound, the fixed discretization approach is wasteful:

1. Thinking of arms in S as "probles": the probes are placed uniformly in the metric space, and in the regret analysis there is a penalty associated with each probe. Whereas it is better to have less probes, as long as the best arm is "covered".

2. Similarity information is discarded once we select the mesh S.

To elaborate on the first point, observe that the discretization error of S is at most the minimal distance between S and the best arm x^*:

$$\mathtt{DE}(S) \le \mathcal{D}(S, x^*) := \min_{x \in S} \mathcal{D}(x, x^*).$$

So it should help to decrease $|S|$ while keeping $\mathcal{D}(S, x^*)$ constant. In particular, if we know that x^* lies in a particular "region" of the metric space, then we do not need to place probes in other regions. Unfortunately, we do not know in advance where x^* is, so we cannot optimize S that way — at least, not in the fixed discretization approach!

However, one may approximately learn the mean rewards of arms over time, and adjust the placement of the probes accordingly, making sure that one has more probes in more "promising" regions of the metric space. This approach is called *adaptive discretization*. Below we describe one implementation of this approach, called the *zooming algorithm*.

What can we hope to improve with this algorithm, given the existence of a matching lower bound? So the goal here is to attain the same worst-case regret as in Theorem 2.6, but do "better" on "nice" problem instances. An important aspect of the overall research challenge here is to quantify the "better" and "nice" in the previous sentence.

The zooming algorithm will bring together three techniques: the "UCB technique" from algorithm UCB1, the new technique of "adaptive discretization", and the "clean event" technique in the analysis.

In what follows, we make two assumptions to simplify presentation:

- There is a known time horizon T.
- The realized rewards can take only finitely many possible values.

Both assumptions can be relaxed with a little more work.

3.1 Algorithm

The algorithm maintains a set $S \subset X$ of "active arms". In each round, some arms may be "activated" according to the "activation rule", and one active arm is selected to according to the "selection rule". Once an arm is activated, it cannot be "deactivated". That is the whole algorithm; we just need to specify the activation rule and the selection rule.

Confidence radius/ball. In each round t, for each active arm x, let $n_t(x)$ is the number of samples for this arm, $\mu_t(x)$ is the reward so far. The confidence radius is defined as

$$r_t(x) = \sqrt{\frac{2 \log T}{n_t(x) + 1}}.$$

The *confidence ball* of arm x is a closed ball in the metric space with center x and radius $r_t(x)$.

$$\mathbf{B}_t(x) = \{y \in X : \mathcal{D}(x,y) \le r_t(x)\}.$$

Activation rule. The activation rule is very simple:

If some arm y becomes uncovered by confidence balls of the active arms, activate y.

The intuition is that for any arm $y \in \mathbf{B}_t(x)$, the algorithm does not have enough samples of x to distinguish x and y, so the two arms look the same, as far as the algorithm is concerned. Therefore the algorithm can sample arm x instead of arm y. So, there is no reason to activate y yet. Going further with this reasoning, we'd like to maintain the invariant:

In each round, all arms are covered by confidence balls of the active arms.

As the algorithm plays some arms over time, their confidence radii (and hence the confidence balls) get smaller, and some arm y may become uncovered. Then we simply activate it! Since immediately after activation the confidence ball of y includes the entire metric space, we see that the invariant is preserved.

Due to the activation rule, the zooming algorithm has the following "self-adjusting property". The algorithm "zooms in" on a given region R of the metric space (i.e., activates many arms in R) if and only if the arms in R are played often. The latter happens (under any reasonable selection rule) if and only if the arms in R have high mean rewards.

Selection rule. We extend the UCB technique from algorithm UCB1. For each active arm x we define an *index* at time t as

$$\texttt{index}_t(x) = \bar{\mu}_t(x) + 2r_t(x)$$

The selection rule is very simple:

Play an active arm with the largest index (break ties arbitrarily).

Recall that algorithm UCB1 chooses an arm with largest UCB on the mean reward, where the UCB is defined as $UCB_t(x) = \bar{\mu}_t(x) + r_t(x)$. So $\texttt{index}_t(x)$ is very similar, and shares the intuition behind UCB1: if $\texttt{index}_t(x)$ is large, then either $\bar{\mu}_t(x)$ is large (so x is a good arm), or $r_t(x)$ is large (so arm x has not been played very often, and should be explored more). And the '+' in the index is a way to trade off exploration and exploitation.

What is new here, compared to UCB1, is that $\texttt{index}_t(x)$ is a UCB not only on the mean reward of x, but also on the mean reward of any arm in the confidence ball of x.

Summary. To sum up the algorithm:

- Maintain a set of active arms S.
- Initially $S = \emptyset$, activate arms one by one.
- In each round t,
  - Activate uncovered arms according to *Activation Rule.*
  - Play the active arm with the largest index $\texttt{index}_t(x)$.

3.2 Analysis: clean event

As in Lecture 2, we define a "clean event" $\mathcal{E}$ and prove that it holds with high probability. Then the rest of the analysis can safely assume that this event holds. The proof that $\mathcal{E}$ holds with high probability is more delicate than the corresponding proofs in Lecture 2, essentially because we cannot immediately take the Union Bound over all of X.

As in Lecture 2, we consider the table of the realized rewards: a row for each arm and T columns, where the j-th column for arm x is the reward for the j-time this arm is chosen by the algorithm. Let us assume, w.l.o.g., that this entire table is chosen before round 1.

We define the *clean event* for arm x as

$$\mathcal{E}_x = \{|\mu_t(x) - \mu(x)| \leq r_t(x), \quad \forall t\}$$

Here, for convenience, we define $\mu_t(x)$ to be 0 if arm x has not yet been played by the algorithm, so that in this case the clean event holds trivially. We are interested in the event $\mathcal{E} = \cap_{x \in X} \mathcal{E}_x$.

Claim 3.1. $\Pr[\mathcal{E}] \geq 1 - \frac{1}{T^2}$.

Proof. By Hoeffding Inequality, $\Pr[\mathcal{E}_x] \geq 1 - \frac{1}{T^4}$ for each arm $x \in X$. However, one cannot immediately apply the Union Bound here because there may be too many arms.

Let X_0 be the set of all arms that can possibly be activated by the algorithm. Note that X_0 is finite; this is because the algorithm is deterministic, the time horizon T is fixed, and, as we assumed upfront, each realized reward can take only finitely many values. (This is the only place where we use that assumption.)

Let N be the total number of arms activated by the algorithm. Denote $[T] := \{1, 2, \ldots, T\}$. Define arms $y_j \in X_0$, $j \in [T]$, as follows

$$y_j = \begin{cases} j\text{-th arm activated}, & \text{if } j \leq N \\ X_N, & \text{otherwise.} \end{cases}$$

Here N and y_j's are random variables, the randomness coming from the reward realizations. Note that $\{y_1, \ldots, y_N\}$ is precisely the set of arms activated in a given execution of the algorithm.

We will prove that $\mathcal{E}_{y_j}$ happens with high probability, for each j. This suffices, because the clean event holds trivially for all arms that are not activated.

Fix an arm $x \in X_0$ and fix $j \in [T]$. Whether or not the event $\{y_j = x\}$ holds is determined by the rewards of other arms. (Indeed, by the time arm x is selected by the algorithm, it is already determined whether x is the j-th arm activated!) Whereas whether or not the clean event $\mathcal{E}_x$ holds is determined by the rewards of arm x alone. It follows that the events $\{y_j = x\}$ and $\mathcal{E}_x$ are independent. Thus:

$$\Pr[\mathcal{E}_{y_j} | y_j = x] = \Pr[\mathcal{E}_x | y_j = x] = \Pr[\mathcal{E}_x] \geq 1 - \tfrac{1}{T^4}$$

Now we can sum over all all $x \in X_0$:

$$\Pr[\mathcal{E}_{y_j}] = \sum_{x \in X_0} \Pr[y_j = x] \cdot \Pr[\mathcal{E}_{y_j} | x = y_j] \geq 1 - \tfrac{1}{T^4}$$

To complete the proof, we apply the Union bound over all $j \in [T]$:

$$\Pr[\mathcal{E}_{y_j}, j \in [T]] \geq 1 - \tfrac{1}{T^3}. \qquad \square$$

We assume the clean event $\mathcal{E}$ from here on.

3.3 Analysis: bad arms

Let us analyze the "bad arms": arms with low mean rewards. We establish two crucial properties: that active bad arms must be far apart in the metric space (Corollary 3.3), and that each "bad" arm cannot be played too often (Corollary 3.4).

Notation. As usual, let $\Delta(x) = \mu^* - \mu(x)$ denote the "badness" of arm x. Let $n(x) = n_{T+1}(x)$ be the total number of samples from arm x.

The following lemma encapsulates a crucial argument which connects the best arm and the arm played in a given round. In particular, we use the main trick from the analysis of UCB1 and the Lipschitz property.

Lemma 3.2. *$\Delta(x) \leq 3\, r_t(x)$ for each arm x and each round t.*

Proof. Fix arm x and round t. Suppose x is played in this round. By the covering invariant, in this round the best arm x^* was covered by the confidence ball of some arm y. It follows that

$$\texttt{index}(x) \geq \texttt{index}(y) = \underbrace{\mu_t(y) + r_t(y)}_{\geq \mu(y)} + r_t(y) \geq \mu(x^*) = \mu^*$$

The last inequality holds because of the Lipschitz condition. On the other hand:

$$\texttt{index}(x) = \underbrace{\mu_t(x)}_{\leq \mu(x) + r_t(x)} + 2 \cdot r_t(x) \leq \mu(x) + 3 \cdot r_t(x)$$

Putting these two equations together: $\Delta(x) := \mu^* - \mu(x) \leq 3 \cdot r_t(x)$.

Now suppose arm x is not played in round t. If it has never been played before round t, then $r_t(x) > 1$ and the lemma follows trivially. Else, letting s be the last time when x has been played before round t, we see that $r_t(x) = r_s(x) \geq \Delta(x)/3$. □

Corollary 3.3. *For any two active arms x, y, we have $\mathcal{D}(x, y) > \frac{1}{3} \min(\Delta(x),\ \Delta(y))$.*

Proof. W.l.o.g. assume that x has been activated before y. Let s be the time when y has been activated. Then $\mathcal{D}(x, y) > r_s(x)$ by the activation rule. And $r_s(x) \geq \Delta(x)/3$ by Lemma 3.2. □

Corollary 3.4. *For each arm x, we have $n(x) \leq O(\log T)\ \Delta^{-2}(x)$.*

Proof. Use Lemma 3.2 for $t = T$, and plug in the definition of the confidence radius. □

3.4 Analysis: regret

For $r > 0$, consider the set of arms whose badness is between r and $2r$:

$$X_r = \{x \in X : r \leq \Delta(x) < 2r\}.$$

Fix $i \in \mathbb{N}$ and let $Y_i = X_r$, where $r = 2^{-i}$. By Corollary 3.3, for any two arms $x, y \in Y_i$, we have $\mathcal{D}(x, y) > r/3$. If we cover Y_i with subsets of diameter $r/3$, then arms x and y cannot lie in the same subset. Since one can cover Y_i with $N_{r/3}(Y_i)$ such subsets, it follows that $|Y_i| \leq N_{r/3}(Y_i)$.

Using Corollary 3.4, we have:

$$R_i(T) := \sum_{x \in Y_i} \Delta(x) \cdot n_t(x) \leq \frac{O(\log T)}{\Delta(x)} \cdot N_{r/3}(Y_i) \leq \frac{O(\log T)}{r} \cdot N_{r/3}(Y_i).$$

Pick $\delta > 0$, and consider arms with $\Delta(\cdot) \leq \delta$ separately from those with $\Delta(\cdot) > \delta$. Note that the total regret from the former cannot exceed δ per round. Therefore:

$$\begin{aligned} R(T) &\leq \delta T + \sum_{i:\, r=2^{-i}>\delta} R_i(T) \\ &\leq \delta T + \sum_{i:\, r=2^{-i}>\delta} \frac{\Theta(\log T)}{r}\, N_{r/3}(Y_i) \\ &\leq \delta T + O(c \cdot \log T) \cdot (\tfrac{1}{\delta})^{d+1}, \end{aligned} \tag{9}$$

where c is a constant and d is some number such that

$$N_{r/3}(X_r) \leq \frac{c}{r^d} \quad \forall r > 0.$$

The smallest (infimum) such d is called the *zooming dimension* with multiplier c.

By choosing $\delta = (\frac{\log T}{T})^{1/(d+2)}$, we obtain

$$R(T) = O\left(T^{\frac{d+1}{d+2}} \, (c \log T)^{\frac{1}{d+2}}\right).$$

Note that we make this chose in the analysis only; the algorithm does not depend on the δ.

Theorem 3.5. *Consider Lipschitz MAB problem with time horizon T. For any given problem instance and any $c > 0$, the zooming algorithm attains regret*

$$\mathbb{E}[R(T)] \leq O\left(T^{\frac{d+1}{d+2}} \, (c \log T)^{\frac{1}{d+2}}\right),$$

where d is the zooming dimension with multiplier c.

While the covering dimension is a property of the metric space, the zooming dimension is a property of the problem instance: it depends not only on the metric space, but on the rewards function μ. In general, the zooming dimension is at most as large as the covering dimension,[2] but may be much smaller. This is because in the definition of the covering dimension one needs to cover all of X, whereas in the definition of the zooming dimension one only needs to cover set X_r

While the regret bound in Theorem 3.5 is appealingly simple, a more precise regret bound is given in (9). Since the algorithm does not depend on δ, this bound holds for all $\delta > 0$.

4 Final remarks

The setting of Lipschitz bandits makes some idealized assumptions: that the Lipschitz condition holds exactly and that it holds with respect to a metric that is fully known to the algorithm. Of course, these assumptions do not necessarily hold in practice.

[2]More precisely: if $d = \texttt{COV}_c$, then the zooming dimension with multiplier $3^d \cdot c$ is at most d.

On the other hand, adaptive discretization is a general technique that may be of use even if the precise assumptions do not hold. To that end, it is worth noting that some of the assumptions in the analysis of the zooming algorithm can be relaxed:

- No need to assume triangle inequality.
- Lipschitz condition needs to hold only for pairs (x^*, y) (for this algorithm), or only in the vicinity of x^* (for another "adaptive discretization" algorithm with essentially the same guarantees).
- Zooming algorithm can be extended to settings in which the similarity between arms is given by a tree / taxonomy on arms (two arms in the same subtree have similar mean rewards), but the algorithm is only given the tree, rather than specific numbers that characterize similarity.

5 Bibliographic notes

Uniform discretization for continuum-armed bandits is from (Kleinberg, 2004); the extension to Lipschitz bandits is from (Kleinberg et al., 2008).

The zooming algorithm is from Kleinberg et al. (2008) (see Kleinberg et al., 2015, for the most recent version). A different algorithm that implements adaptive discretization, with very similar regret bounds, appeared in the follow-up paper (Bubeck et al., 2011).

The lower bound in Section 1.2 has been proved in (Kleinberg, 2004), in a much stronger formulation and a much more complicated proof. The present proof is unpublished.

For a discussion of the work on Lipschitz bandits and related problems, see "related work" section in (Kleinberg et al., 2015).

References

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. Online Optimization in X-Armed Bandits. *J. of Machine Learning Research (JMLR)*, 12:1587–1627, 2011.

Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *18th Advances in Neural Information Processing Systems (NIPS)*, 2004.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *40th ACM Symp. on Theory of Computing (STOC)*, pages 681–690, 2008.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. Working paper, published at `http://arxiv.org/abs/1312.1277`, 2015. Merged and revised version of conference papers in *ACM STOC 2008* and *ACM-SIAM SODA 2010*.