

Introduction to Multi-Armed Bandits

From Aleksandrs Slivkins
Microsoft Research NYC

20183160 Kim Tae Hyeon

Contents

| | |
|--|-----------|
| 1 Bandits with IID Rewards (rev. Jul'18) | 1 |
| 1.1 Model and examples | 1 |
| 1.2 Simple algorithms: uniform exploration | 3 |
| 1.3 Advanced algorithms: adaptive exploration | 5 |
| 1.4 Bibliographic remarks and further directions | 10 |
| 1.5 Exercises and Hints | 12 |
| 2 Lower Bounds (rev. Jul'18) | 15 |
| 2.1 Background on KL-divergence | 16 |
| 2.2 A simple example: flipping one coin | 19 |
| 2.3 Flipping several coins: "bandits with prediction" | 20 |
| 2.4 Proof of Lemma 2.10 for $K \geq 24$ arms | 22 |
| 2.5 Instance-dependent lower bounds (without proofs) | 23 |
| 2.6 Bibliographic remarks and further directions | 24 |
| 2.7 Exercises and Hints | 26 |
| Interlude A: Bandits with Initial Information (rev. Jan'17) | 27 |
| 3 Thompson Sampling (rev. Jan'17) | 29 |
| 3.1 Bayesian bandits: preliminaries and notation | 29 |
| 3.2 Thompson Sampling: definition and characterizations | 30 |
| 3.3 Computational aspects | 31 |
| 3.4 Example: 0-1 rewards and Beta priors | 32 |
| 3.5 Example: Gaussian rewards and Gaussian priors | 33 |
| 3.6 Bayesian regret | 34 |
| 3.7 Thompson Sampling with no prior (and no proofs) | 37 |
| 3.8 Bibliographic remarks and further directions | 37 |
| 4 Lipschitz Bandits (rev. Jul'18) | 39 |
| 4.1 Continuum-armed bandits | 39 |
| 4.2 Lipschitz MAB | 43 |
| 4.3 Adaptive discretization: the Zooming Algorithm | 45 |
| 4.4 Bibliographic remarks and further directions | 50 |
| 4.5 Exercises and Hints | 53 |
| 5 Full Feedback and Adversarial Costs (rev. Sep'17) | 55 |

| | |
|---|------------|
| 5.1 Adversaries and regret | 56 |
| 5.2 Initial results: binary prediction with experts advice | 58 |
| 5.3 Hedge Algorithm | 60 |
| 5.4 Bibliographic remarks and further directions | 64 |
| 5.5 Exercises and Hints | 64 |
| 6 Adversarial Bandits (rev. Jun'18) | 67 |
| 6.1 Reduction from bandit feedback to full feedback | 68 |
| 6.2 Adversarial bandits with expert advice | 68 |
| 6.3 Preliminary analysis: unbiased estimates | 69 |
| 6.4 Algorithm Exp4 and crude analysis | 70 |
| 6.5 Improved analysis of Exp4 | 71 |
| 6.6 Bibliographic remarks and further directions | 73 |
| 6.7 Exercises and Hints | 74 |
| 7 Linear Costs and Combinatorial Actions (rev. Jun'18) | 77 |
| 7.1 Bandits-to-experts reduction, revisited | 77 |
| 7.2 Online routing problem | 78 |
| 7.3 Combinatorial semi-bandits | 80 |
| 7.4 Follow the Perturbed Leader | 82 |
| 8 Contextual Bandits (rev. Jul'18) | 87 |
| 8.1 Warm-up: small number of contexts | 88 |
| 8.2 Lipschitz contextual bandits | 88 |
| 8.3 Linear contextual bandits: LinUCB algorithm (no proofs) | 90 |
| 8.4 Contextual bandits with a policy class | 91 |
| 8.5 Bibliographic remarks and further directions | 94 |
| 8.6 Exercises and Hints | 95 |
| 9 Bandits with Knapsacks (rev. May'18) | 97 |
| 9.1 Definitions, examples, and discussion | 97 |
| 9.2 Groundwork: fractional relaxation and confidence regions | 101 |
| 9.3 Three algorithms for BwK (no proofs) | 102 |
| 9.4 Bibliographic remarks and further directions | 104 |
| 10 Bandits and Zero-Sum Games (rev. May'17) | 105 |
| 10.1 Basics: guaranteed minimax value | 106 |
| 10.2 Convergence to Nash Equilibrium | 107 |
| 10.3 Beyond zero-sum games: coarse correlated equilibrium | 110 |
| Bibliography | 113 |

Where to use

1. News

Problem a user visits a news site, the site presents it with a news header, and a user either clicks on this header or not.

Goal to maximize the number of clicks

Assumption

each user is drawn independently from a fixed distribution over users

In each round, the click happens independently with a probability that depends only on the chosen header.

2. Ad selection

Problem for a user, if ad a is displayed, the website observes whether the user clicks on the ad, in which case the advertiser pays some amount $v_a \in [0,1]$

Goal to maximize the paid amount (ad = arm, reward = paid amount)

Assumption

The paid amount v_a depends only on the displayed ad, but does not change over time.

Introduction

Problem

Problem protocol: Multi-armed bandits

In each round $t \in [T]$:

1. Algorithm picks arm $a_t \in \mathcal{A}$.
2. Algorithm observes reward $r_t \in [0, 1]$ for the chosen arm.

1. The restriction to the interval $[0, 1]$
2. The reward for each action is i.i.d. (independent and identically distributed).
3. Reward distribution is the Bernoulli distribution.

$$R(t) = \mu^* \cdot t - \sum_{s=1}^t \mu(a_s)$$

Definition of Regret at round t.

$R(t)$ is a random variable, so we should talk about expected regret $E[R(T)]$.

Algorithm

1. Uniform exploration
2. Adaptive exploration
 - 2.1. Successive elimination algorithm
 - 2.2. UCB1 algorithm
 - 2.3. Thompson sampling

Uniform exploration

- 1 Exploration phase: try each arm N times;
- 2 Select the arm \hat{a} with the highest average reward (break ties arbitrarily);
- 3 Exploitation phase: play arm \hat{a} in all remaining rounds.

Algorithm 1.1: Explore-First with parameter N .

$$N = (T/K)^{2/3} \cdot O(\log T)^{1/3} \quad \text{Put this to } N$$

Theorem 1.3. *Explore-first achieves regret $\mathbb{E}[R(T)] \leq T^{2/3} \times O(K \log T)^{1/3}$, where K is the number of arms.*

Its performance in the exploration phase is terrible.

Uniform exploration & epsilon greedy

```
1 for each round  $t = 1, 2, \dots$  do  
2   | Toss a coin with success probability  $\epsilon_t$ ;  
3   | if success then  
4   |   | explore: choose an arm uniformly at random  
5   | else  
6   |   | exploit: choose the arm with the highest average reward so far  
7 end
```

Algorithm 1.2: Epsilon-Greedy with exploration probabilities $(\epsilon_1, \epsilon_2, \dots)$.

Theorem 1.4. *Epsilon-greedy algorithm with exploration probabilities $\epsilon_t = t^{-1/3} \cdot (K \log t)^{1/3}$ achieves regret bound $\mathbb{E}[R(t)] \leq t^{2/3} \cdot O(K \log t)^{1/3}$ for each round t .*

It is the same regret as in uniform exploration, but it holds for all rounds t .
To get (epsilon t) ϵ_t from $\epsilon_t \sim t^{-1/3}$

Uniform exploration

With epsilon greedy

Both exploration-first and epsilon-greedy have a big flaw that the exploration schedule does not depend on the history of the observed rewards

Adaptive exploration

Successive Elimination Algorithm

- 1 Alternate two arms until $UCB_t(a) < LCB_t(a')$ after some even round t ;
- 2 Then abandon arm a , and use arm a' forever since.

Algorithm 1.3: “High-confidence elimination” algorithm for two arms

This approach extends to $K > 2$ arms as follows:

- 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. $\exists \text{arm } a' \text{ with } UCB_t(a) < LCB_t(a')$;
- 5 Repeat until end of rounds.

Algorithm 1.4: Successive Elimination algorithm

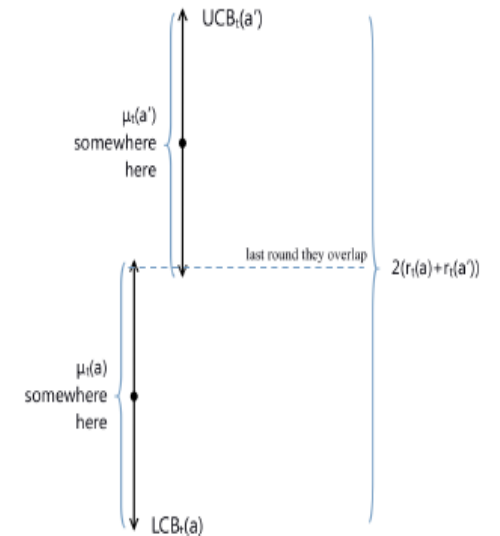


Figure 1.2: t is the last round that the two confidence intervals still overlap

Adaptive exploration

Successive Elimination Algorithm

Lemma 1.6. *For two arms, Algorithm 1.3 achieves regret $\mathbb{E}[R(t)] \leq O(\sqrt{t \log T})$ for each round $t \leq T$.*

Theorem 1.8. *Successive Elimination algorithm achieves regret*

$$\mathbb{E}[R(t)] = O(\sqrt{Kt \log T}) \quad \text{for all rounds } t \leq T.$$

Theorem 1.9. *Successive Elimination algorithm achieves regret*

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

The existence of logarithmic regret bounds is benefit of adaptive exploration compared to non-adaptive exploration.

Adaptive exploration

UCB1 Algorithm

- 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 1.5: UCB1 Algorithm

$$r_t(a) = \sqrt{\frac{\alpha \cdot \ln t}{n_t(a)}}$$

Original version of
confidence radius

Theorem 1.14. *Algorithm UCB1 satisfies regret bounds in (1.9) and (1.11).*

$$\mathbb{E}[R(t)] = O(\sqrt{Kt \log T}) \quad \text{for all rounds } t \leq T. \quad (1.9)$$

$$\mathbb{E}[R(T)] \leq O(\log T) \left[\sum_{\text{arms } a \text{ with } \mu(a) < \mu(a^*)} \frac{1}{\mu(a^*) - \mu(a)} \right]. \quad (1.11)$$

Adaptive exploration

A regret bound of the form $C * f(T)$, where $f()$ does not depend on the mean rewards μ , and the “constant” C does not depend on T .

Instance-independent

C does not depend on T

Instance-dependent

Otherwise

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

Instance independent

Theorem 2.14. Fix K , the number of arms. Consider an algorithm such that

$$\mathbb{E}[R(t)] \leq O(C_{\mathcal{I}, \alpha} t^\alpha) \quad \text{for each problem instance } \mathcal{I} \text{ and each } \alpha > 0. \quad (2.15)$$

Here the “constant” $C_{\mathcal{I}, \alpha}$ can depend on the problem instance \mathcal{I} and the α , but not on time t .

Fix an arbitrary problem instance \mathcal{I} . For this problem instance:

$$\text{There exists time } t_0 \text{ such that for any } t \geq t_0 \quad \mathbb{E}[R(t)] \geq C_{\mathcal{I}} \ln(t), \quad (2.16)$$

for some constant $C_{\mathcal{I}}$ that depends on the problem instance, but not on time t .

Theorem 2.16. For each problem instance \mathcal{I} and any algorithm that satisfies (2.15),

(a) the bound (2.16) holds with

$$C_{\mathcal{I}} = \sum_{a: \Delta(a) > 0} \frac{\mu^*(1 - \mu^*)}{\Delta(a)}.$$

(b) for each $\epsilon > 0$, the bound (2.16) holds with

$$C_{\mathcal{I}} = \sum_{a: \Delta(a) > 0} \frac{\Delta(a)}{\text{KL}(\mu(a), \mu^*)} - \epsilon.$$

Adaptive exploration

Thompson Sampling

Theorem 3.12. Consider IID bandits with no priors. For Thompson Sampling with both approaches (i) and (ii) we have: $\mathbb{E}[R(T)] \leq \mathcal{O}(\sqrt{kT \log T})$.

Theorem 3.13. Consider IID bandits with no priors. For Thompson sampling with approach (i),

$$\mathbb{E}[R(T)] \leq (1 + \epsilon)(\log T) \underbrace{\sum_{\substack{\text{arms } a \\ s.t. \Delta(a) > 0}} \frac{\Delta(a)}{KL(\mu(a), \mu^*)}}_{(*)} + \frac{f(\mu)}{\epsilon^2},$$

for all $\epsilon > 0$. Here $f(\mu)$ depends on the reward function μ , but not on the ϵ , and $\Delta(a) = \mu(a^*) - \mu(a)$.

Prior work considered two such “fake priors”:

- (i) independent, uniform priors and 0-1 rewards,
- (ii) independent, Gaussian priors and Gaussian unit-variance rewards (so each reward is distributed as $\mathcal{N}(\mu(a), 1)$, where $\mu(a)$ is the mean).

Main definition. For each round t , consider the posterior distribution for the best arm a^* . Formally, it is distribution p_t over arms given by

$$p_t(a) = \mathbb{P}[a = a^* | H_t] \quad \text{for each arm } a. \quad (3.1)$$

Thompson Sampling is a very simple algorithm:

$$\text{In each round } t, \text{ arm } a_t \text{ is drawn independently from distribution } p_t. \quad (3.2)$$

Sometimes we will write $p_t(a) = p_t(a|H_t)$ to emphasize the dependence on history H_t .

Alternative characterization. Thompson Sampling can be stated differently: in each round t ,

1. sample reward function μ_t from the posterior distribution $\mathbb{P}_t(\mu) = \mathbb{P}(\mu|H_t)$.
2. choose the best arm \tilde{a}_t according to μ_t .

Let us prove that this characterization is in fact equivalent to the original algorithm.

Lemma 3.1. For each round t and each history H_t , arms a_t and \tilde{a}_t are identically distributed.

Proof. For each arm a we have:

$$\begin{aligned} \Pr(\tilde{a}_t = a) &= \mathbb{P}_t(\text{arm } a \text{ is the best arm}) && \text{by definition of } \tilde{a}_t \\ &= \mathbb{P}(\text{arm } a \text{ is the best arm} | H_t) && \text{by definition of the posterior } \mathbb{P}_t \\ &= p_t(a|H_t) && \text{by definition of } p_t. \end{aligned}$$

Thus, \tilde{a}_t is distributed according to distribution $p_t(a|H_t)$. □