DDA4230 Reinforcement learning ε-greedy and ETC

Lecture 5

Lecturer: Baoxiang Wang *Scribe: Baoxiang Wang*

1 Goal of this lecture

To understand the ε-greedy algorithm and the explore-then-commit algorithm and complete our first analysis with a logarithmic regret.
Suggested reading: Chapter 6 of *Bandit Algorithms*;

2 Recap: Multi-armed bandits

The problem of multi armed bandits is a special case of the MDP we defined

- $\mathcal{A} = [m] = \{1, 2, \dots, m\}$;
- $\mathcal{R}(s, a) = r(a)$ some unknown stochastic function $r(\cdot)$

with a finite horizon T. The policy is aware of the problem structure but has no prior knowledge of the reward function. We mainly consider the asymptotic worst-case performance, namely, if the algorithm achieves a regret of either $O(\log T)$, $O(T)$, or some other orders, and the constant associated.

The policy $\pi(t)$ is a mapping from the current time t to an action. We define the regret as

$$\overline{R}_t = \sum_{i=1}^{m} \mathbb{E}[N_{t,i}]\Delta_i,$$

where $N_{t,i} = \mathbb{E}[\sum_{t'=0}^{t} \mathbb{1}\{a_{t'} = i\}]$ and $\Delta_i = \mu^* - \mu_i$.

In the following analysis, without loss of generality we assume that arm 1 is optimal. From the form $\overline{R}_t$ is written above, it is intuitive to bound $N_{t,i}$ for each i to analyze the performance of a given policy.

We use $x \in \arg\max\{\dots\}$ to denote the set of optimums of a given term. When an algorithm breaks ties in $\arg\max$ arbitrarily, we write $x = \arg\max\{\dots\}$ for simplicity.

Denote $\Delta_{\min} = \min_{i:\Delta_i>0} \Delta_i$ as the minimum non-zero gap between an arm and an optimal arm.

3 Greedy-based algorithms

3.1 The greedy algorithm

The idea of the greedy algorithm is to pull each arm once and then always pull the arm with the best empirical mean reward. This algorithm focuses purely on exploitation and does not consider exploration.

The worst-case regret of the greedy algorithm is $O(T)$. In fact, any algorithm achieves a regret at most $O(T)$. It suffices to show that the greedy algorithms obtain this regret in some bandit instances. Consider a two-armed bandit instance where $r(1)$ and $r(2)$ are Bernoulli distribution with mean p and q respectively (assume $p > q$, without loss of generality), then $\mathbb{P}(r_1 = 0, r_2 = 1) = q(1-p)$. When this event is true, the algorithm will pull arm 2 for the rest of the horizon, which induces a regret of at least $q(1-p)\Delta_2 T - o(T)$.

Algorithm 1: The greedy algorithm

Output: $\pi(t), t \in \{0, 1, \ldots, T\}$

while $1 \le t \le m$ **do**

$$\pi(t) = t$$

while $m < t \le T$ **do**

$$\pi(t) = \arg\max_{i \in [m]} \left\{ \frac{1}{N_{t-1,i}} \sum_{t'=0}^{t-1} r_{t'} \mathbb{1}\{a_{t'} = i\} \right\}$$

3.2 The ε-greedy algorithm

A variant of the greedy algorithm, which is built upon the philosophy of being optimistic is good, is derived to include exploration in the algorithm. The ε-greedy algorithm takes a non-deterministic policy that forces exploration on arms which look sub-optimal. The details are given below.

Algorithm 2: The ε-greedy algorithm

Input: $\varepsilon_t, t \in \{0, 1, \ldots, T\}$ the exploration parameters

Output: $\pi(t), t \in \{0, 1, \ldots, T\}$

while $1 \le t \le m$ **do**

$$\pi(t) = t$$

while $m < t \le T$ **do**

$$\pi(t) \sim \begin{cases} \arg\max_{i \in [m]} \left\{ \frac{1}{N_{t-1,i}} \sum_{t'=0}^{t-1} r_{t'} \mathbb{1}\{a_{t'} = i\} \right\} & \text{with probability } 1 - \varepsilon_t \\ i \quad \text{with probability } \varepsilon_t / m, \text{ for each } i \in [m] \end{cases}$$

The algorithm amounts to the choice of the exploration parameters ε_t.

We first establish a negative result when ε_t does not diminish with t. In fact, if $\varepsilon_t > \varepsilon$ holds for some constant $\varepsilon > 0$, then for $T - m$ rounds, the algorithm has a probability at least ε to pull a random arm. As pulling a random arm induces an expected regret of $\frac{1}{m}(\Delta_2 + \cdots + \Delta_m)$, the regret of the algorithm is at least

$$\overline{R}_t = \frac{1}{m}(\Delta_2 + \cdots + \Delta_m)\varepsilon T + o(T).$$

Again, a regret in order $O(T)$ is usually not desired.

By carefully choosing ε_t as a decreasing function of t, we can obtain an algorithm with its regret at most $O(\log T)$.

Theorem 1 *Assume that $r(i)$ is 1-subgaussian for each i. Let and $\varepsilon_t = \min\{1, Ct^{-1}\Delta_{\min}^{-2}m\}$, where C is some sufficiently large absolute constant. Then, the regret under the ε-Greedy algorithm satisfies*

$$\overline{R}_T \leq C' \sum_{i \geq 2} \left(\Delta_i + \frac{\Delta_i}{\Delta_{\min}^2} \log \left\{ e, \frac{T\Delta_{\min}^2}{m} \right\} \right),$$

where C' is an absolute constant.

The proof of the theorem is two-fold. First, the cost of exploration, being $\overline{R}_t = \frac{1}{m}(\Delta_2 + \cdots + \Delta_m)\varepsilon T$ for $\varepsilon_t = O(1)$, reduces to $\overline{R}_t = \frac{1}{m}(\Delta_2 + \cdots + \Delta_m)(1 + \frac{1}{2} + \cdots + \frac{1}{T}) = \frac{1}{m}(\Delta_2 + \cdots + \Delta_m)\log T$ with the annealing of ε_t. Second, we show that the probability of pulling a suboptimal arm for more than $\log T$ times is very thin (as thin as at most $O(T^{-1})$). This can be done by showing that the empirical mean of a suboptimal gap has small enough probability to deviate by at least Δ_i, compared to the empirical mean of the optimal arm established by at least $\log t/m$ pulls on average.

ε-greedy, with Theorem 1, is the first algorithm we introduce to obtain a logarithmic regret. Despite this, the choice for ε, which is horizon-independent, requires information on the smallest suboptimality gap and an unspecified constant C. The performance of the algorithm can be uncertain in practice.

4 Explore-then-commit algorithms

Different from greedy algorithms, the Explore-then-Commit (ETC) algorithm assigns multiple rounds of exploration in the beginning of the algorithm and commits to the identified best arm without exploration thereafter. With careful choice of the number of rounds of exploration the algorithm also achieves an asymptotic regret of $\log T$.

The idea behind Algorithm 3 is simple. Divide n rounds into two parts, including the first mk rounds for equal exploration on each arm and the remaining time for exploitation on the arm demonstrating the best performance in the exploration period. We must settle for a trade-off between these two parts. If k is too small, there is a considerable chance that the exploration is poor, resulting in the exploitation procedure being suboptimal. If k is too large, the regret generated in the exploration process will likely dominate. Therefore, the best k is usually set at some balanced point.

Algorithm 3: The explore-then-commit algorithm

Input: k: number of exploration on each arm.
Output: $\pi(t), t \in \{0, 1, \ldots, T\}$
while $t \leq km$ **do**

$$a_t = \lceil t \bmod m \rceil$$

while $km < t \leq n$ **do**

$$a_t = \arg\max_{i \in [m]} \frac{1}{k} \sum_{t'=0}^{mk} r_{t'} \mathbb{1}\{a_{t'} = i\}$$

Theorem 2 *Assume that $r(i)$ is 1-subgaussian for each i. The regret under ETC satisfies*

$$\overline{R}_T \leq k \sum_{i \in [m]} \Delta_i + (T - mk) \sum_{i \in [m]} \Delta_i e^{-k\Delta_i^2/4}. \tag{1}$$

Particularly, for two-armed bandits ($m = 2$), taking $k = \max\left\{1, 4\Delta_2^{-2} \log(T\Delta_2^2/4)\right\}$ yields

$$\overline{R}_T \leq \Delta_2 + (4 + e^{-2})\sqrt{T}. \tag{2}$$

Regret bounds depending on Δ_i, like (1), are called gap dependent (also known as problem dependent, instance dependent, and distribution dependent). The reason is that the algorithm need a specification of the bandit instance as part of its parameters. Regret bounds that depend only on T are gap independent (also known as problem independent and problem free).

Theorem 2 follows from a direct application of concentration inequalities. There are a few things worth remarking here. First, a high-probability version of the result on the pseudo-regret $\overline{R}_T$ can be obtained similarly:

$$\mathbb{P}\left(\bar{R}_T = T\mu_1 - \sum_{t \in [T]} \mu_{\pi(t)} \leq k \sum_{i \in [m]} \Delta_i\right) \geq 1 - \sum_{i \in [m]} e^{-k\Delta_i^2/4}.$$

Second, despite the fact that (2) gives an optimal bound on regret (for which we will prove it later), how to achieve it depends on knowledge of both the suboptimality gaps Δ_i and the horizon n. These quantities are usually fixed but may not be revealed to the agent in advance. In theory, it can be shown that for two-armed bandits the dependence on Δ_i can be removed while obtaining a sub-optimal regret bound $T^{2/3}$, and the dependence on n can be resolved by a doubling trick without increasing the regret by too much.

Acknowledgement

This lecture notes partially use material from *Reinforcement learning: An introduction*, and *Bandit algorithms*. We also used some scribe from *Stochastic Bandits* by user Yiminithere on GitHub. Proofread by Shaokui Wei.