Lecture 4 - Greedy algorithms

Guiliang Liu

The Chinese University of Hong Kong, Shenzhen

DDA4230: Reinforcement Learning
Course Page: [Click]

DDA 4230 Resources

Please join our Slack group.

https://join.slack.com/t/slack-us51977/shared_invite/zt-22g8b40v8-0qSs9o0G3~8hXHwWydlCpw

Please check our course page.

https://guiliang.github.io/courses/cuhk-dda-4230/dda_4230.html

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Greedy algorithms

Greedy Algorithm: 1) pull each arm once and then 2) always pull the arm with the best empirical mean reward.

Algorithm 1: The greedy algorithm

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

while $0 \leq t \leq m-1$ **do**

$$\pi(t) = t + 1$$

while $m \leq t \leq T$ **do**

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

The Regret of Greedy algorithms

Consider a two-armed bandit instance where $r(1)$ and $r(2)$ follow Bernoulli distributions with mean p and q respectively.

- If the event $(r_1 = 0, r_2 = 1)$ (with probability $q(1-p)$) is true, the algorithm will pull arm 2 for the rest of the horizon.
- induce a regret of at least $q(1-p)\Delta_2 T + o(T)$.

The worst-case regret of the greedy algorithm is $O(T)$ (Note $O(T)$ is the worst).

ε Greedy algorithms

ε-greedy algorithm: takes a non-deterministic policy that forces exploration on sub-optimal arms. which is built upon the philosophy of being optimistic is good.

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 $0 \le t \le m-1$ **do**

$$\pi(t) = t + 1$$

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

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

深 圳）
Hong Kong, Shenzhen

The Regret of ε Greedy algorithms

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

- 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 $\frac{1}{m}(\Delta_2 + \cdots + \Delta_m)$ per step (arm 1 is the best, so $\Delta_1 = 0$), the regret of the algorithm is at least:

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

The worst-case regret of the greedy algorithm is $O(T)$.

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

The Regret of ε Greedy algorithms

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

- 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

Assume that $r(i)$ is 1-sub-Gaussian for each i. By choosing $\varepsilon_t = \min\{1, Ct^{-1}\Delta_{\min}^{-2} m\}$ for some sufficiently large constant C, the regret under the ε-greedy algorithm satisfies

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

where C' is an absolute constant.

Proof Schema

The proof of the theorem is two-fold.

- The cost of exploration, being $\overline{R}_t = \frac{1}{m}(\Delta_2 + \cdots + \Delta_m)\varepsilon$ for $\varepsilon_t = O(1)$, reduces to $\overline{R}_t = \frac{1}{m}(\Delta_2 + \cdots + \Delta_m)O(1 + \frac{1}{2} + \cdots + \frac{1}{T}) = \frac{1}{m}(\Delta_2 + \cdots + \Delta_m)O(\log T)$[1] with the annealing of ε_t.
- Show that the probability of pulling a suboptimal arm in a round after $\log T$ explorations is very thin (as thin as at most $O(\log T/T)$).

[1] The nth partial sum of the harmonic series, $H_n = 1 + 1/2 + 1/3 + \ldots + 1/n$, is approximately $\log(n)$

Some Remarks of ε Greedy algorithms

Remarks of the Theorem:

- ε-greedy algorithm is the first algorithm we introduce to obtain a logarithmic regret (this is in fact the best regret).
- The choice for ε requires information on the gap of suboptimality.

Without prior knowledge, one has to pull each arm for a few times to get an estimation of this gap and plug in the estimation (known as bootstrap).

Question and Answering (Q&A)