Econ 425 Week 10 Part I

Reinforcement learning 强化学习

Grigory Franguridi

UCLA Econ

USC CESR

franguri@usc.edu

Motivation

- problem: every morning, choose a coffee place
- if you **stick to one coffee place** (exploitation), you are missing out on the coffee served by competitors
- however, if you **try different coffee places one by one** (exploration), the probability of encountering the worst coffee of your life is high!
- on the other hand, chances to find better coffee

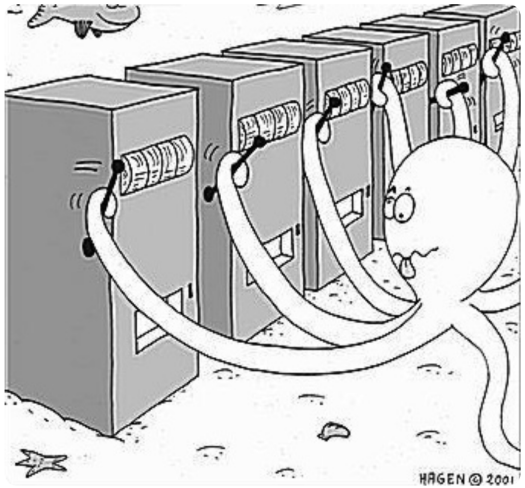
Motivation

- dilemma arises from **incomplete information**: we need to gather enough information by exploring new actions to formulate the best overall strategy
- eventually leads to minimizing the overall bad experiences
- **multi-armed bandit** is a mathematical representation
  - used in online advertising, e-commerce, A/B testing, healthcare, finance, etc.

Multi-armed bandit problem (MABP)

- bandit is someone who steals your money
- 1-armed bandit is a slot machine wherein you insert a coin, pull a lever, and get a reward
  - all casinos configure these slot machines in such a way that all gamblers end up losing money
- multi-armed bandit is a slot machine
  with several levers with different rewards
- not only reward is random, but **even distribution of reward is unknown** to gambler

Multi-armed bandit problem (MABP)



- **goal**: identify which lever to pull to get maximum reward after a given set of trials
- choosing an arm = *action* $\rightarrow$ *reward*
- mathematically, a single-step Markov decision process

Bernoulli MABP

- Bernoulli: reward $\in \{0, 1\}$
- sample results for a 5-armed Bernoulli bandit:

Arm	Reward
1	0
2	0
3	1
4	1
5	0
3	1
3	1
2	0
1	1
4	0
2	0

Bernoulli MABP

- looks like arm 3 gives the maximum return and hence one idea is to keep playing this arm to obtain the maximum reward (pure exploitation)
- arm 5 might look like a bad arm to play, but we have played this arm only once; maybe should play it a few more times (exploration) to gain info
- only then decide which arm to play (exploitation)

MABP: applications

- **Online advertising** maximize ad revenue
- advertiser makes revenue every time an ad is clicked
- exploration (collect information on ad's performance using click-through rates) vs exploitation (stick with ad that has performed best so far)

MABP: applications

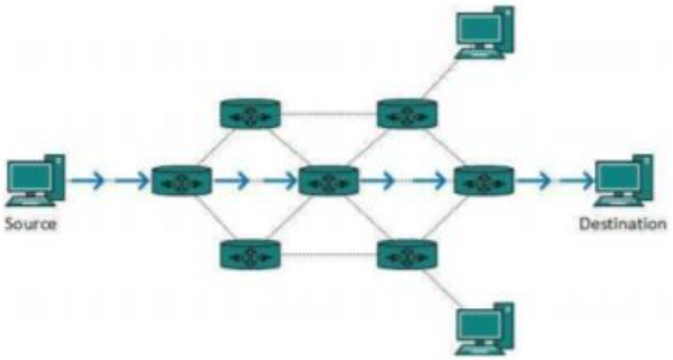
Clinical trials: the well-being of patients is as important as the results of the study

Exploration = identifying the best treatment

Exploitation = treating patients most effectively during the trial

MABP: applications

- **Network routing**: process of selecting a path for traffic in a network (telephone/computer networks, internet)
- allocation of channels to users maximizing the overall throughput can be formulated as a MABP



MABP: applications

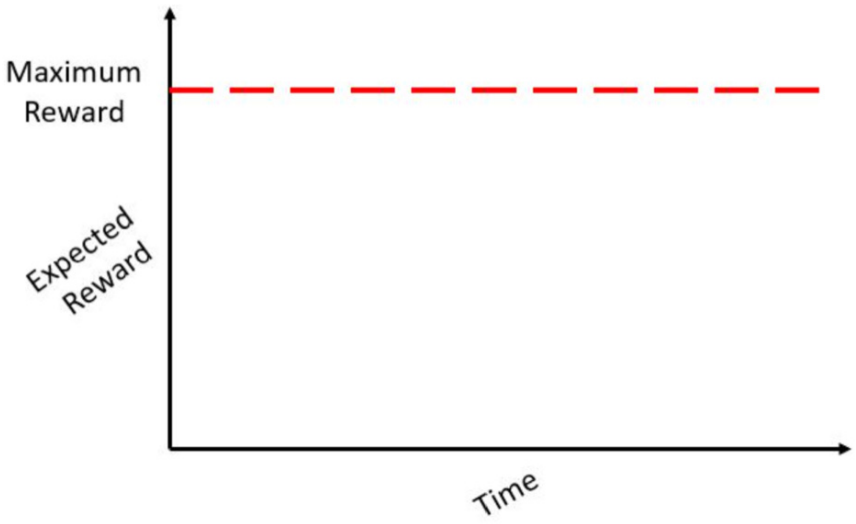
Game design: test experimental changes in gameplay/UI and exploit the changes delivering positive experiences for players

Solution strategies

- **value function**: $Q_t(a) = \mathbb{E}[r|a]$, where r is reward, a is action, t is time
- $Q_t(a_k) = p_k$, where $p_1, \ldots, p_K$ are the reward probabilities for a K-armed bandit
- how to evaluate a given strategy?
- one way is to compare the total/average reward after n trials
- another way is **regret**

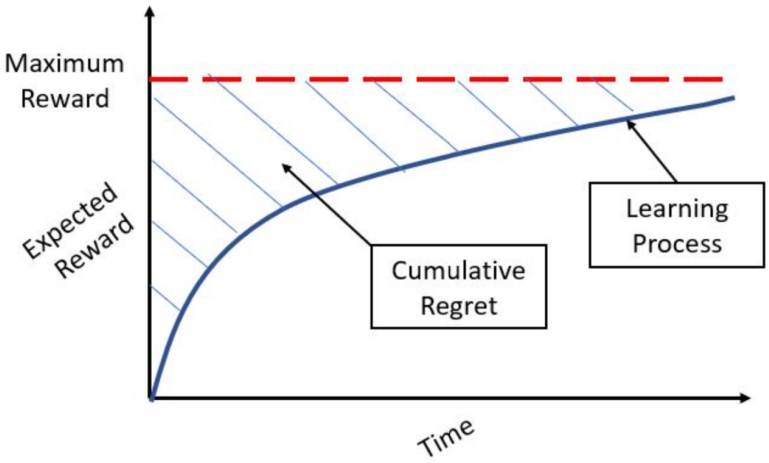
Regret

if pull the best arm repeatedly, get a maximum *expected* reward (horizontal line):



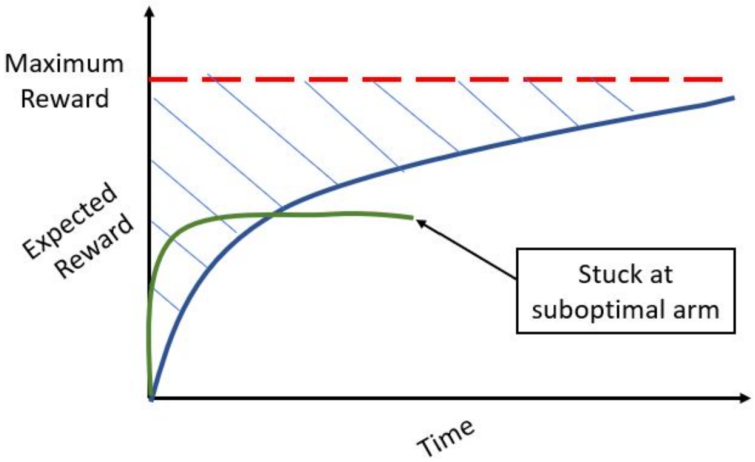
Regret

- in the real world, unsure about which arm is best
- loss incurred during exploitation is called **regret** (from not choosing the best arm)



Regret

- how does regret change if not enough exploration / too much exploitation of a suboptimal arm?



No exploration (greedy approach)

- naive approach: choose the best arm so far

$$Q_t(a) = \frac{\sum_{i=1}^{t} 1\{a_t = a\} \cdot r_i}{\sum_{i=1}^{t} 1\{a_t = a\}}$$

$$a^* = \text{argmax}_{a \in \mathcal{A}} Q_t(a)$$

No exploration (greedy approach)

- need the entire history of rewards
- avoid this by using running sum:

$$Q_t(a) = \frac{Q_{t-1}(a)N_t(a_t) + R_t \cdot 1\{a_t = a\}}{N_t(a_t)}$$

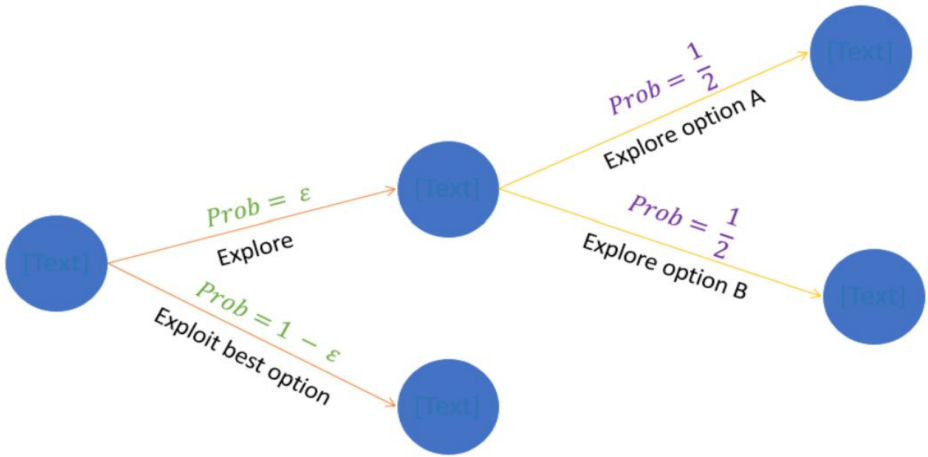
and

$$Q_t(a) = Q_{t-1}(a) + \frac{1}{N_t(a_t)(R_t - Q_{t-1}(a))}$$

- this approach **never explores**, as it always picks the same arm

Epsilon-greedy approach

- w. probability ε, choose a random action (exploration)
- w. probability 1-ϵ, choose an action maximizing $Q_t(a)$
- example w. two actions A and B

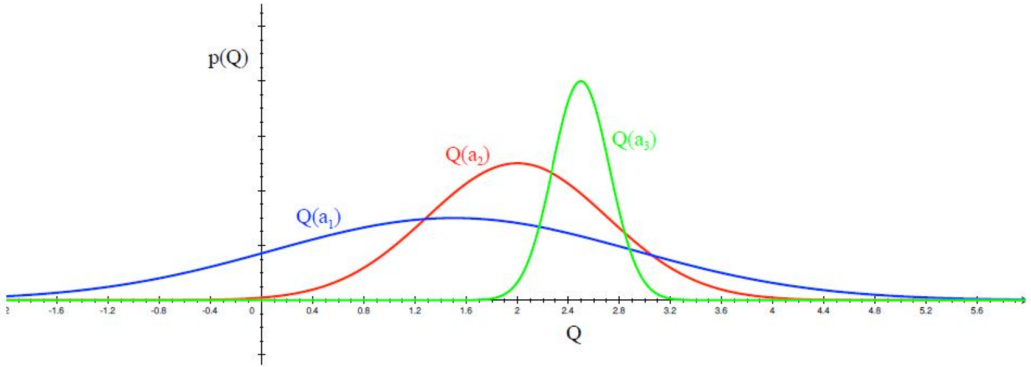


Upper Confidence Bound

- the most widely used solution method for MABP
- based on the principle of **optimism in the face of uncertainty**: the more uncertainty about an arm, the more important to explore it

Upper Confidence Bound

Distribution of reward for 3 arms after a few trials:



- reward for a_1 has the highest variance (maximal uncertainty)
- UCB:
  - choose a_1 and receive a reward
  - if still uncertain about a_1, choose it again until the uncertainty is below a threshold

Upper Confidence Bound

Intuition: at each time, two cases

1. **optimism is justified**: get a positive reward
2. **optimism is not justified**: play an arm that we falsely believe gives a large reward.
    - if this happens sufficiently often, then we learn the payoff distribution and stop choosing this arm in the future

Upper Confidence Bound

- UCB is a family of algorithms, focus on **UCB1**:
  1. for $t = 1, \ldots, K$: play arm t to obtain initial value of $Q(a_t)$
  2. for $t > K + 1$: play

$$a^* = \arg\max_{a \in \mathcal{A}} \left\{ Q_t(a) + \sqrt{\frac{2 \log t}{N_t(a)}} \right\},$$

  where $N_t(a)$ is the number of times action a was played so far, and update $Q_t(a^*)$

Regret comparison

- among the algorithms discussed so far, only the UCB algorithm has regret increasing as $\log(t)$
- other algorithms have linear regret with different slopes