Lecture 09: Exploration strategies in RL

Nikolay Karpachev

Acknowledgements

The following great sources were used to build this lecture:

- David Silver’s lecture on exploration
- Lecture from Practical RL YSDA course

Outline

- Exploration vs. Exploitation tradeoff in RL
- Multi-armed bandits
- Exploration strategies
    - Simple heuristic-based
    - “Optimism in the face of uncertainty”
    - Probability matching

Exploration vs. Exploitation in RL

- Online decision-making involves a fundamental choice
  - **Exploitation:** Make the best decision given the current information
  - **Exploration:** Gather more information
- The best long-term strategy may involve short-term sacrifices
- Agent should gather enough relevant information to make reasonable decisions

Exploration vs. Exploitation: examples

- Restaurant selection
  - **Exploitation**: Go to your favourite restaurant
  - **Exploration**: Try new restaurant
- Online banner advertisements
  - **Exploitation**: Show the most successful advert
  - **Exploration**: Show a different advert
- Game playing
  - **Exploitation**: Play the move you believe is the best
  - **Exploration**: Play a different move

Multi-armed bandit

- What is a bandit?

Multi-armed bandit

Multi-armed bandit

- A single state
- Set of possible actions (decide which slot machine to play)
- Each machine has an unknown probability of success
- Goal: maximize the total number of successful games

Regret

$$Q(a) = \mathbb{E}\left[r|a\right]$$

$$V^* = Q(a^*) = \max_{a \in \mathcal{A}} Q(a)$$

- Regret (Total Regret): opportunity loss for one step (all steps)

$$l_t = \mathbb{E}\left[V^* - Q(a_t)\right]$$

$$L_t = \mathbb{E}\left[\sum_{\tau=1}^{t} V^* - Q(a_\tau)\right]$$

We want to minimize the total regret

Exploration strategies so far

- Eps-greedy
  - With p = eps, take random action. Optimal otherwise
- Boltzman (aka softmax)
  - Pick actions proportionately to scaled Q-values

$$P(a) = softmax\left(\frac{Q(s,a)}{std}\right)$$

- Decaying eps-greedy
  - Same as eps-greedy; start with high eps, decrease it during training

Greedy algorithm

- Always selects actions with highest values
- What is the total regret?

Greedy algorithm

- Always selects actions with highest values
- What is the total regret?

- Greedy can lock to a suboptimal action forever
- Hence, **linear total regret**

Epsilon-greedy algorithm

- Explores forever
- Selects suboptimal actions with fixed probability over and over again
- Linear total regret

Epsilon-greedy with decay

- Has a decay schedule for eps
- With properly selected schedule, has a **logarithmic total regret**
- But to design a proper schedule can be tricky

Optimism in the face of uncertainty

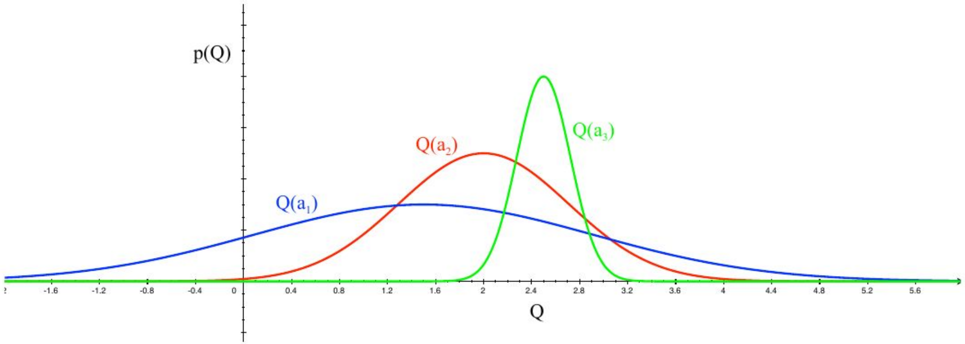
- How do humans explore?
- For example, which of the following questions would you like to investigate?
  - Whether humans can fly by pulling their hair up
  - Whether the new cafe next to the office serves good breakfast

Optimism in the face of uncertainty

- How do humans explore?
- For example, which of the following questions would you like to investigate:
  - Whether humans can fly by pulling their hair up
  - Whether the new cafe next to the office serves good breakfast

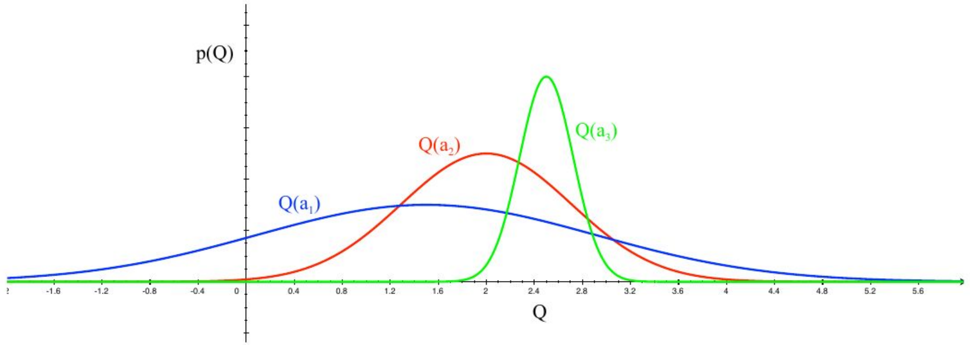
- We want to try actions if we believe there’s a chance they are good

Optimism in the face of uncertainty



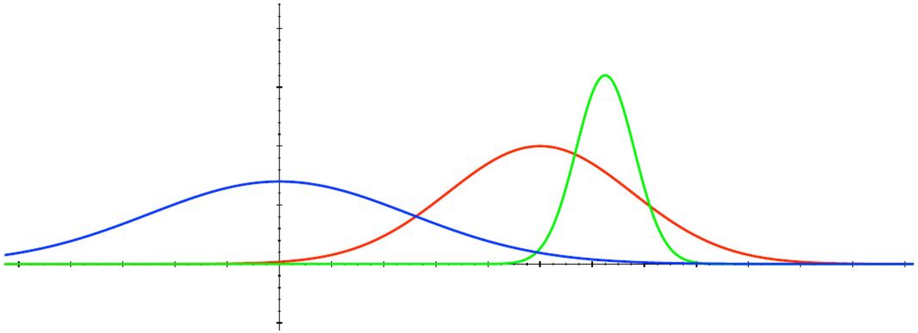
- Which action should we pick?

Optimism in the face of uncertainty



- Which action should we pick?
- The more uncertain we are about an action value
- The more important it is to try that action
- It could turn out to be the best action

Optimism in the face of uncertainty



- After picking blue action
- We are less uncertain about the value
- And more likely to pick another action

Upper confidence bounds

- We want to select
  - Uncertain outcomes
  - With greater expected value

Upper confidence bounds

- We want to select
  - Uncertain outcomes
  - With greater expected value

- Let’s compute 95% upper confidence bound for each action
- Take action with the highest upper confidence bound

Upper confidence bounds

Theorem (Hoeffding's inequality):

Given a sample of a random variable bounded in [0, 1],

$$\mathbb{P}\left[\mathbb{E}\left[X\right] > \overline{X}_t + u\right] \leq e^{-2tu^2}$$

Upper confidence bounds

- We can apply Hoeffding's inequality to the case of bandits:

$$\mathbb{P}\left[Q(a) > \hat{Q}_t(a) + U_t(a)\right] \leq e^{-2N_t(a)U_t(a)^2}$$

$$e^{-2N_t(a)U_t(a)^2} = p$$

$$U_t(a) = \sqrt{\frac{-\log p}{2N_t(a)}}$$

Upper confidence bounds

- With fixed p (e.g. 95% UCB)

$$U_t(a) = \sqrt{\frac{-\log p}{2N_t(a)}}$$

- Possible extension: reduce p during training (ucb converges to 1 in the limit; this guarantees optimal solution)

$$p = t^{-4} \qquad\qquad U_t(a) = \sqrt{\frac{2 \log t}{N_t(a)}}$$

UCB-1 algorithm

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\; Q(a) + \sqrt{\frac{2 \log t}{N_t(a)}}$$

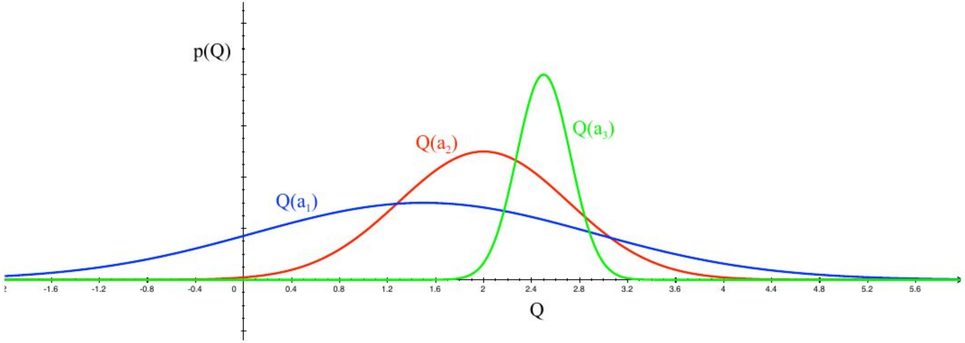
- Achieves **logarithmic total regret**

Bayesian UCB

- Assign prior distribution P(Q(s,a))
- Learn posterior P(Q(s,a)|data)
- Take q-th percentile of P(Q(s,a)) and select the best action

Probability matching

- Select action a according to the probability that **a is the optimal action**



$$\pi(a \mid h_t) = \mathbb{P}\left[Q(a) > Q(a'), \forall a' \neq a \mid h_t\right]$$

Thompson sampling

- Compute posterior distribution for each Q(s,a)
- Sample from each action’s posterior
- Select action with max value on sample
- **Thompson sampling will select action proportionately to the probability that this action is optimal**

Outro

This lecture covered:

- Exploration-vs-exploitation tradeoff in RL
- How to compare exploration strategies
- Algorithms:
  - Greedy, eps-greedy, softmax-sampling
  - Upper confidence bound based sampling
  - Probability matching and thompson sampling

Thanks for the attention