Advanced Topics in Reinforcement Learning

Lecture 2: Bandits

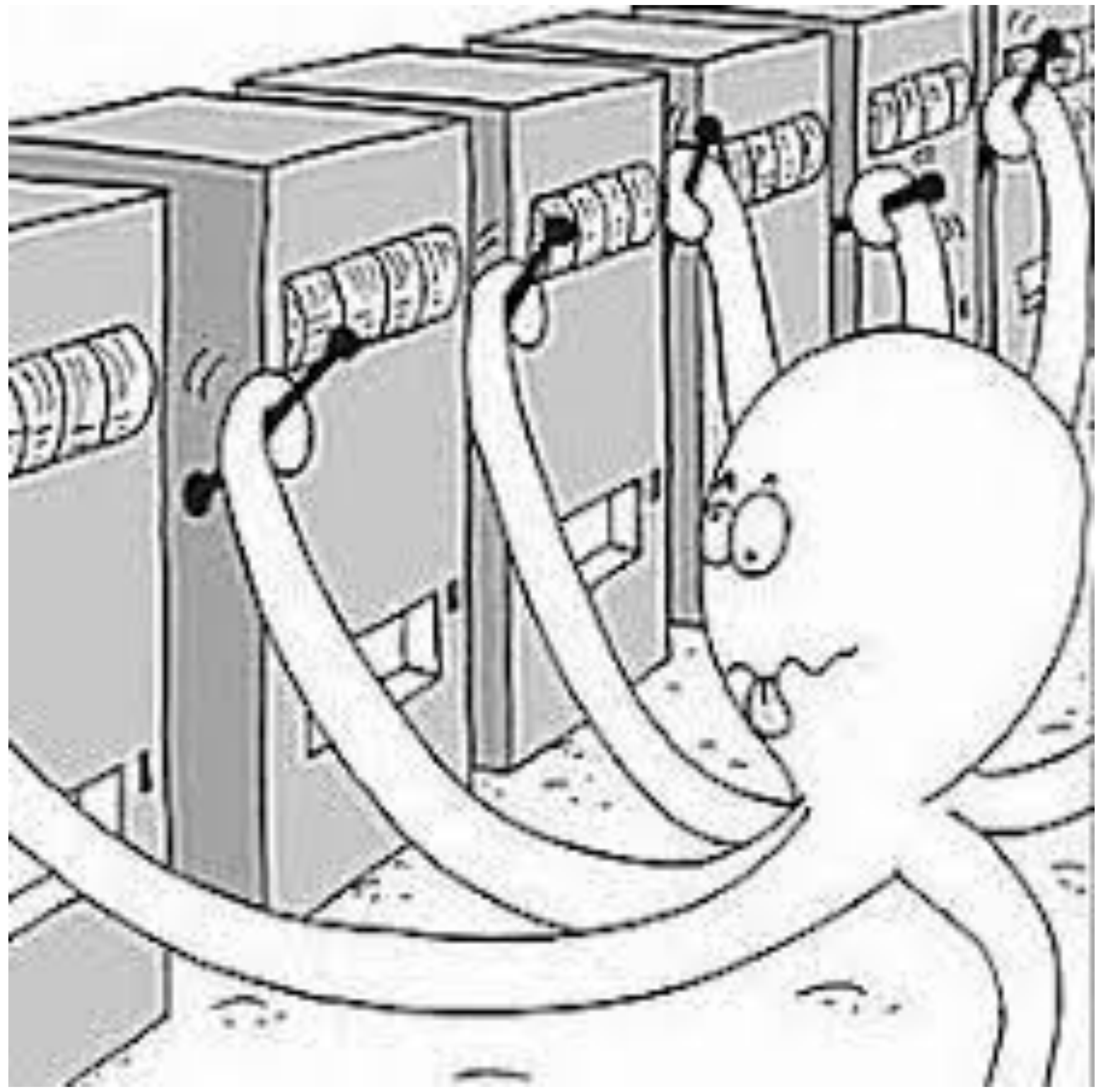
Josiah Hanna
University of Wisconsin — Madison

Announcements

- Good job on reading responses
    - Examples
- Reading Sign-Ups: https://docs.google.com/spreadsheets/d/1-dce7-qzt8EVM4gYOLIl5WzYEGpioWM4x0VyA6QimzY/edit#gid=0

Why Study Bandits?

- Simplest model of sequential decision-making.
- Build intuition for important concepts; many concepts extend to the more complex decision processes we focus on in this course.
- Today’s lecture scratches the surface of a deep topic with extensive research, many applications, and many variations.
  - https://tor-lattimore.com/downloads/book/book.pdf



Today's Outline

- Do I have a bandit problem?
- Estimating action-values.
- Exploration vs. Exploitation.
- Policy-based learning.

General Reinforcement Learning

- States: $s \in \mathcal{S}$
- Actions: $a \in \mathcal{A}$
- Rewards: $R \sim r(s, a)$
- State transitions: $S \sim P(s, a)$
- Goal: Find a policy, $\pi : \mathcal{S} \rightarrow \mathcal{A}$, that maximizes cumulative reward.

Bandit Problems

- ~~States: $s \in \mathcal{S}$~~ No state (or equivalently $|\mathcal{S}| = 1$)
- Actions: $a \in \mathcal{A}$ (also called "arms")
- ~~Rewards: $R \sim r(s, a)$~~ $R \sim r(a)$ with expected value $q(a)$.
- ~~State transitions: $S \sim P(s, a)$~~ Actions do not affect future decisions.
- ~~Goal: Find a policy, $\pi : \mathcal{S} \rightarrow \mathcal{A}$, that maximizes cumulative reward.~~
- Goal: Find the action with highest expected reward.

Bandit Interaction

- Choose an action, observe a reward, repeat.
- $a_0, r_0, a_1, r_1, \ldots$
- Find highest reward arm as quickly as possible.
- Measure algorithm performance with regret:

$$R_T = \sum_{t=0}^{T} q(a^\star) - q(a_t)$$

Attractions and Limitations

- Bandits are a simple model that requires solving explore-exploit trade-off.
- Widely applicable to real world problems.
- No state — take an action and immediately faced with the same situation.
    - No need for planning or reasoning about delayed rewards.
- Immediate pay-off for action choice.
    - No need for credit assignment

Is my application a bandit?

- Hyper-parameter optimization for machine learning.
- Recommend ads and web content.
- Recommend medical treatments.
- Sending push notifications to promote app engagement.

Action-Values

- Need to estimate expected rewards for each arm. Denote the estimate as $Q_t(a)$.
- Each time we pull an arm, we update $Q_t(a)$.
- $Q_t(a) = \dfrac{\sum_{i=0}^{t} R_t \cdot I\{A_t = a\}}{N_t(a)}$.
- $Q_t(a)$ is the estimated *action-value* for action a at time t.
- $A_{t+1} \leftarrow \arg\max_{a \in \mathscr{A}} Q_t(a)$

Action-Values

- What is time and memory complexity of action-value updates?

$$O(t)$$

- $Q_t(a) = \frac{\sum_{i=0}^{t} R_t \cdot I\{A_t = a\}}{N_t(a)} = Q_{t-1}(a) + \frac{1}{N_t(a)}(R_t - Q_{t-1}(a))$
- New Estimate <— Old Estimate + Step Size * (Target - Old Estimate)

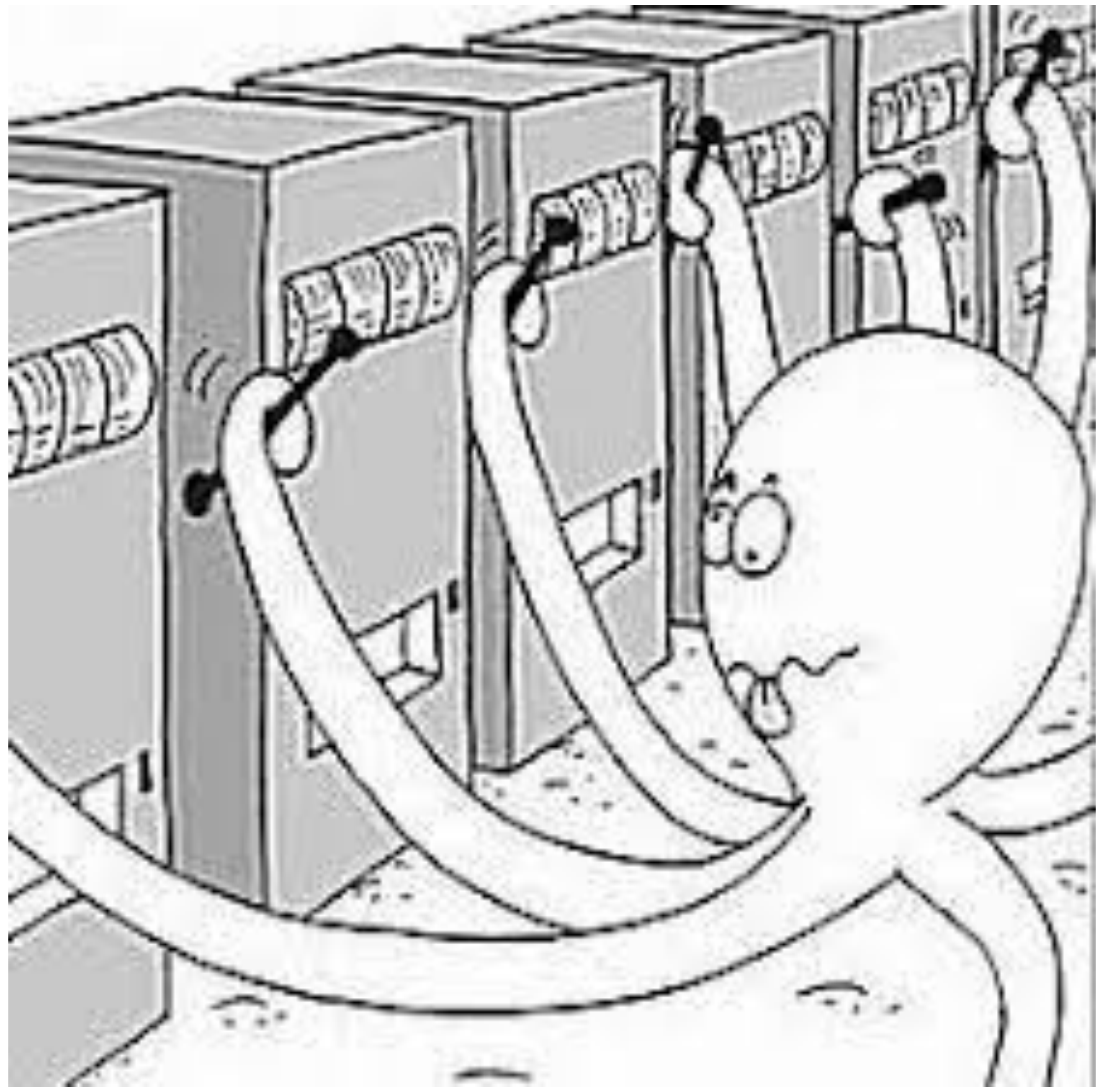
Step-Size Selection

- How does the choice of step-size affect algorithm behavior?
- New Estimate <— Old Estimate + Step Size * (Target - Old Estimate)
- When might you want a big step size? Small step-size?

Exploration vs Exploitation

“..the problem [exploration-exploitation] was proposed [by British scientist] to be dropped over Germany so that German scientists could also waste their time on it.”

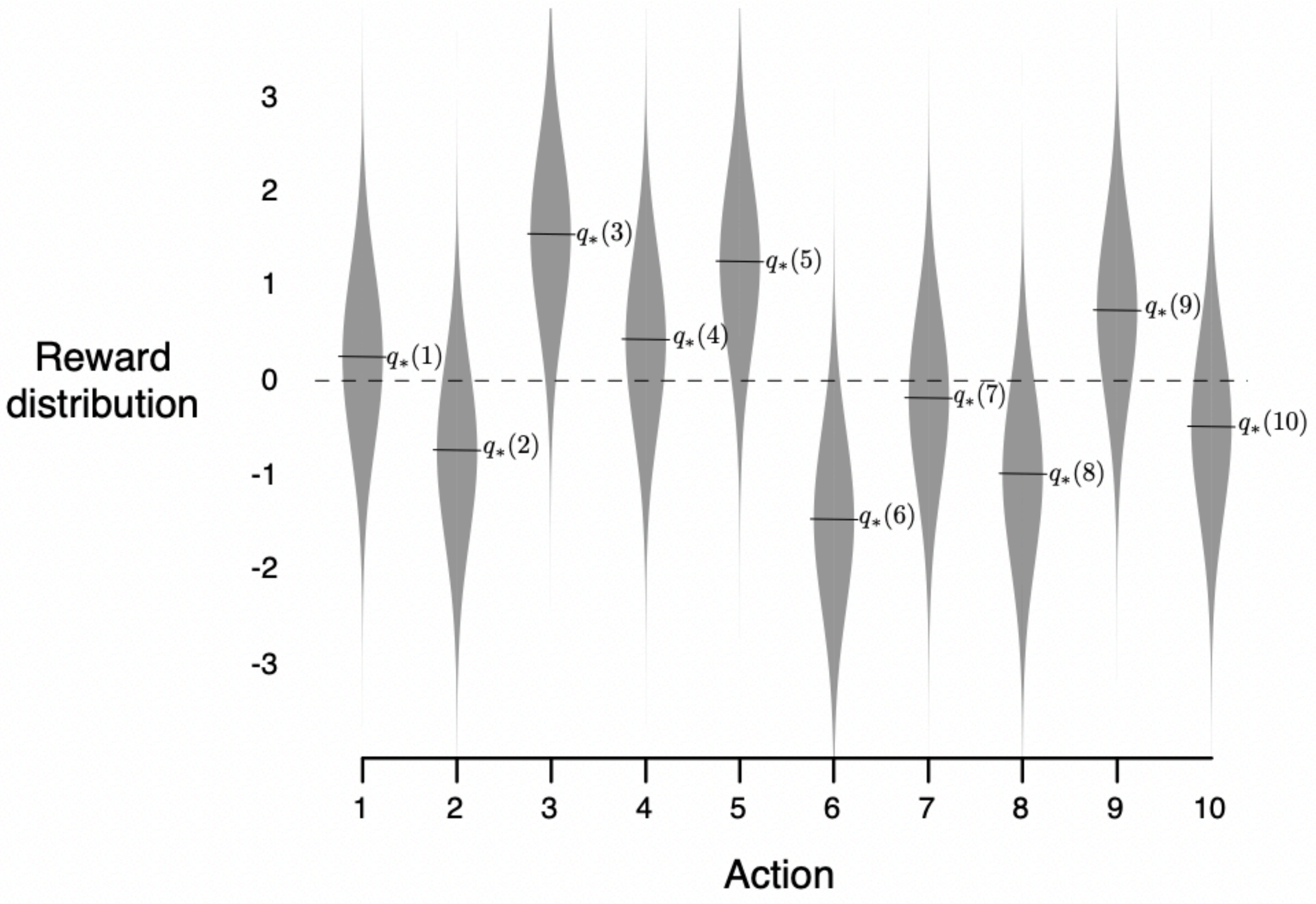
- Peter Whittle



Naive Exploration

- Greedy action selection: $A_{t+1} \leftarrow \arg\max_{a \in \mathcal{A}} Q_t(a)$
- What might go wrong?
- Simple Solution: pull the arm with the best **estimated** reward with probability $1 - \epsilon$, otherwise pull a random arm.
- The value of ϵ controls how much exploration we do.

Naive Exploration



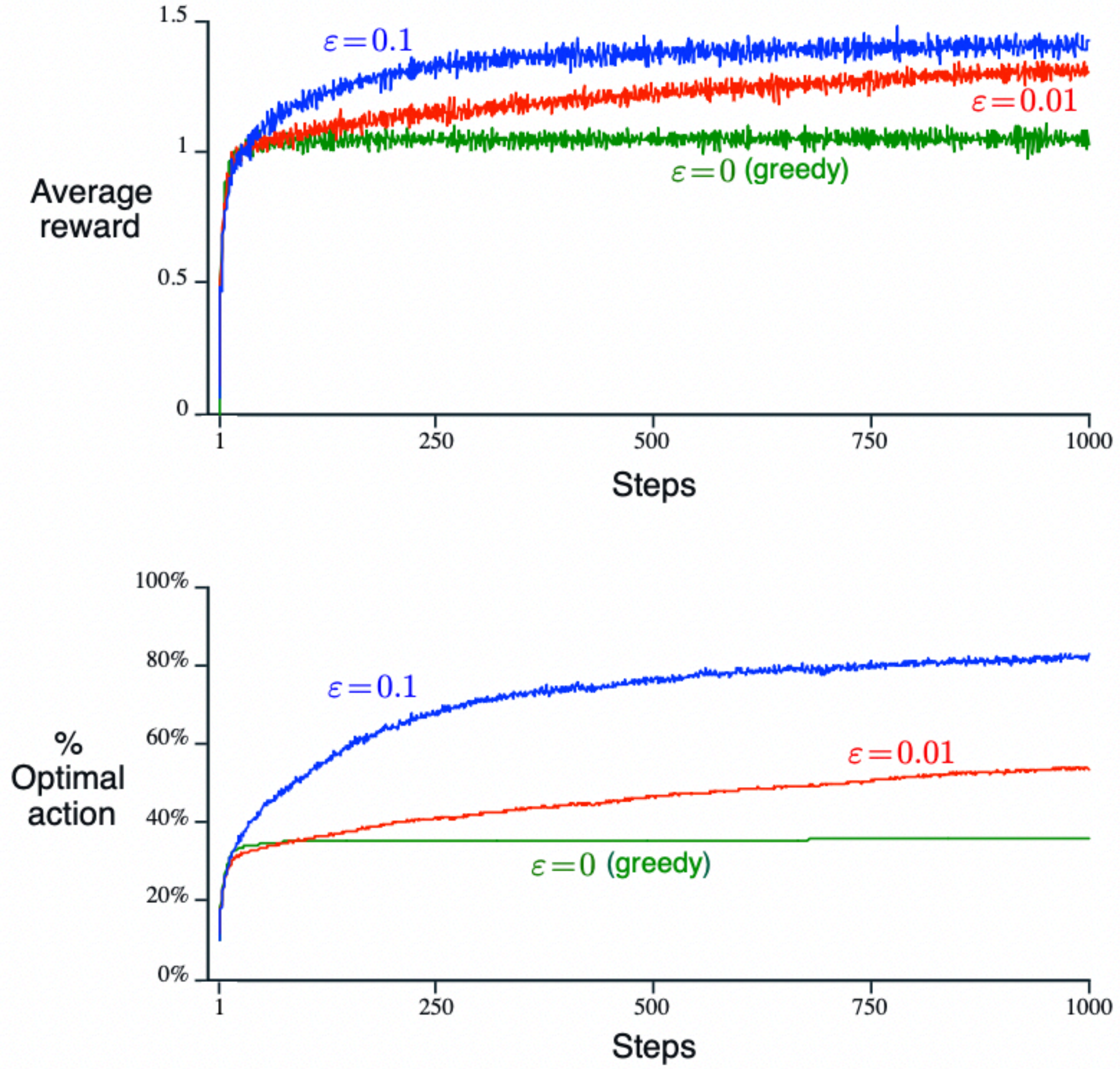


Figure 2.2: Average performance of ε-greedy action-value methods on the 10-armed testbed. These data are averages over 2000 runs with different bandit problems. All methods used sample averages as their action-value estimates.

Optimistic Initialization

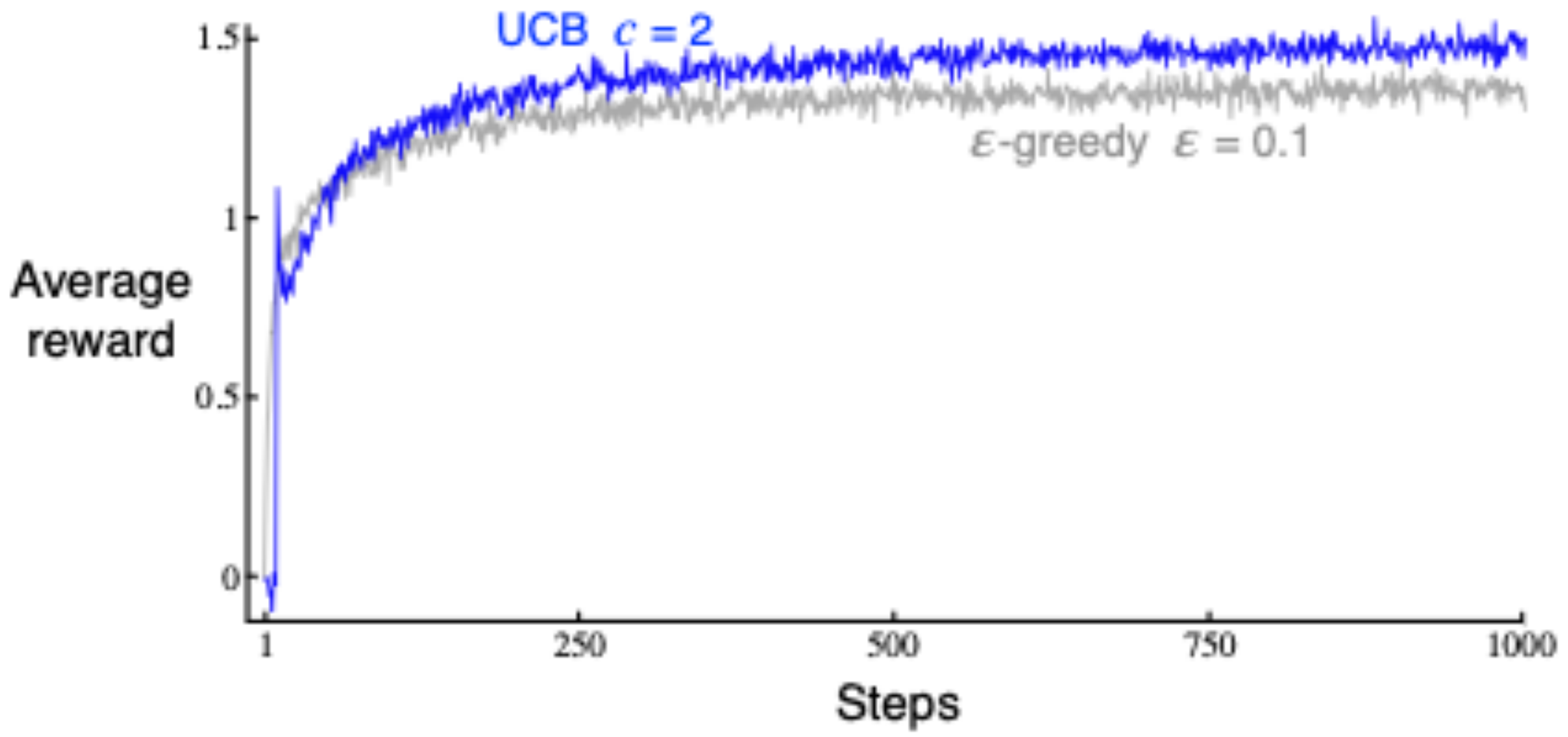
- Set initial values, $Q_0(a)$, to high values for all arms. Then epsilon-greedy will favor untried arms.
- Simple heuristic to improve exploration.
- Injects an initial bias into action-value estimates.
- Too much attention to the initial step?

UCB: Advanced Exploration

- Epsilon-greedy never stops exploring arms **even after clearly sub-optimal.**

- The upper confidence bound algorithm only explores actions that could have the highest expected reward.

- $$A_{t+1} = \arg\max Q_t(a) + c\sqrt{\frac{\ln(t)}{N_t(a)}}$$



- The parameter c controls exploration vs. exploitation.

Policy-based Learning

- Ultimately, we just want action selection!
- Instead of estimating action-values, maintain action preferences.
- Let $H_t(a)$ be the preference for action a at time t.
- Select action with softmax probability: $\Pr(A_{t+1} = a) := \frac{\exp H_t(a)}{\sum_b \exp H_t(b)} = \pi(a)$
- Update preferences:

$$H_{t+1}(A_t) \leftarrow H_t(A_t) + \alpha(R_t - \overline{R}_t)(1 - \pi_t(A_t))$$

$$H_{t+1}(a) \leftarrow H_t(a) - \alpha(R_t - \overline{R}_t)\pi_t(a) \qquad a \neq A_t$$

Learning with a baseline

Follow the book's derivation but don't include the baseline B or equivalently set it to zero!

- Basic update rule: $H_{t+1}(A_t) \leftarrow H_t(A_t) + \alpha R_t(1 - \pi_t(A_t))$

- How does this update change the preferences? What happens when all rewards are positive? All negative?

- With a baseline: $H_{t+1}(A_t) \leftarrow H_t(A_t) + \alpha(R_t - \overline{R}_t)(1 - \pi_t(A_t))$

Update in proportion to how much better than average instead of how much above zero.

- Value of $\overline{R}_t$ doesn't change expected update **as long as independent of** A_t**.**

Contextual Bandit Problems

- States: $s \in \mathcal{S}$ Now we have multiple states.
- Actions: $a \in \mathcal{A}$ (still called “arms”)
- Rewards: $R \sim r(s, a)$ with expected value $q(s, a)$.
- ~~State transitions: $S \sim P(s, a)$~~ Actions do not affect future state probability.
- Goal: Find a policy, $\pi : \mathcal{S} \rightarrow \mathcal{A}$, that maximizes cumulative reward.

Summary

- Multi-armed bandits provide a simplified setting for studying sequential decision-making.
- Estimating action-values provides a means to optimal action selection.
- Must sufficiently explore to find maximum reward action.
- A policy can be learned directly without estimating action-values.
- Key ideas: action-values, incremental updates, step-sizes, epsilon-greedy exploration, gradient-based learning.

Action Items

- Join Piazza!
- Read and send responses if haven't.
- Presentation sign-ups (posted on Piazza).