

10703 Deep Reinforcement Learning

Exploration vs. Exploitation

Tom Mitchell

October 22, 2018

Reading: Barto & Sutton, Chapter 2

Used Materials

- Some of the material and slides for this lecture were taken from Chapter 2 of Barto & Sutton textbook.
- Some slides are borrowed from Ruslan Salakhutdinov and Katerina Fragkiadaki, who in turn borrowed from Rich Sutton's RL class and David Silver's Deep RL tutorial

Exploration vs. Exploitation Dilemma

- ▶ Online decision-making involves a fundamental choice:
 - **Exploitation**: Take the most rewarding action given current knowledge
 - **Exploration**: Take an action to gather more knowledge
- ▶ The best long-term strategy may involve **short-term sacrifices**
- ▶ Gather enough knowledge early to make the best long term decisions

Exploration vs. Exploitation Dilemma

- ▶ Restaurant Selection
 - **Exploitation**: Go to your favorite restaurant
 - **Exploration**: Try a new restaurant
- ▶ Oil Drilling
 - **Exploitation**: Drill at the best known location
 - **Exploration**: Drill at a new location
- ▶ Game Playing
 - **Exploitation**: Play the move you believe is best
 - **Exploration**: Play an experimental move

Exploration vs. Exploitation Dilemma

- ▶ **Naive Exploration**
 - Add noise to greedy policy (e.g. ϵ -greedy)
- ▶ **Optimistic Initialization**
 - Assume the best until proven otherwise
- ▶ **Optimism in the Face of Uncertainty**
 - Prefer actions with uncertain values
- ▶ **Probability Matching**
 - Select actions according to probability they are best
- ▶ **Information State Search**
 - Look-ahead search incorporating value of information

The Multi-Armed Bandit

- ▶ A multi-armed bandit is a tuple $\langle A, R \rangle$
- ▶ A is a known set of k actions (or “arms”)
- ▶ $\mathcal{R}^a(r) = \mathbb{P}[r|a]$ is an unknown probability distribution over rewards, given actions
- ▶ At each step t the agent selects an action $a_t \in \mathcal{A}$
- ▶ The environment generates a reward $r_t \sim \mathcal{R}^{a_t}$
- ▶ **The goal** is to maximize cumulative reward $\sum_{\tau=1}^t r_{\tau}$
- ▶ What is the best strategy?



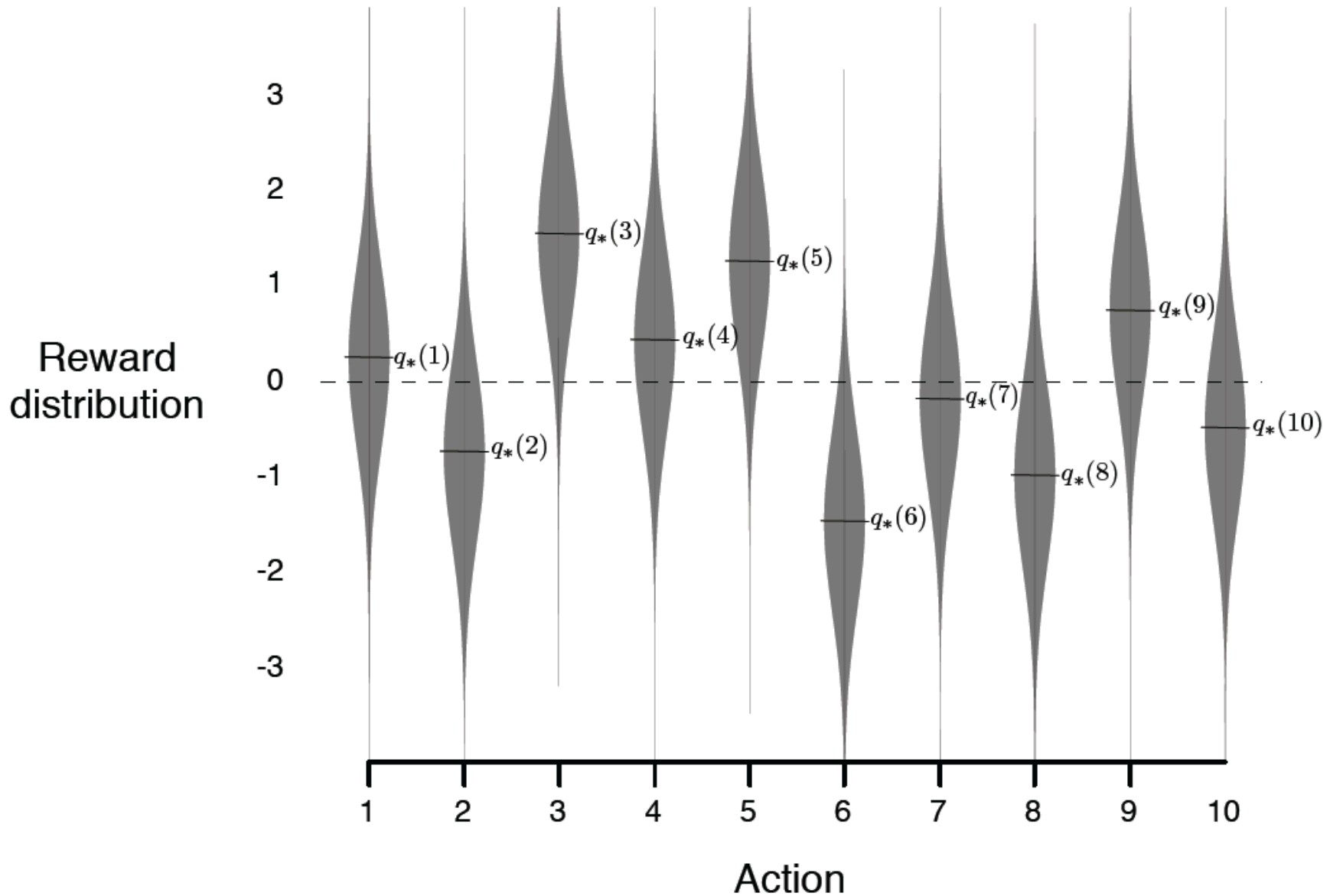


Figure 2.1: An example bandit problem from the 10-armed testbed. The true value $q_*(a)$ of each of the ten actions was selected according to a normal distribution with mean zero and unit variance, and then the actual rewards were selected according to a mean $q_*(a)$ unit variance normal distribution, as suggested by these gray distributions.

Regret

- ▶ The action-value is the mean (i.e. expected) reward for action a ,

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

- ▶ The optimal value V^* is

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

- ▶ The regret is the expected opportunity loss for one step

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

- ▶ The total regret is the opportunity loss summed over steps

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

- ▶ Maximize cumulative reward = minimize total regret

Counting Regret

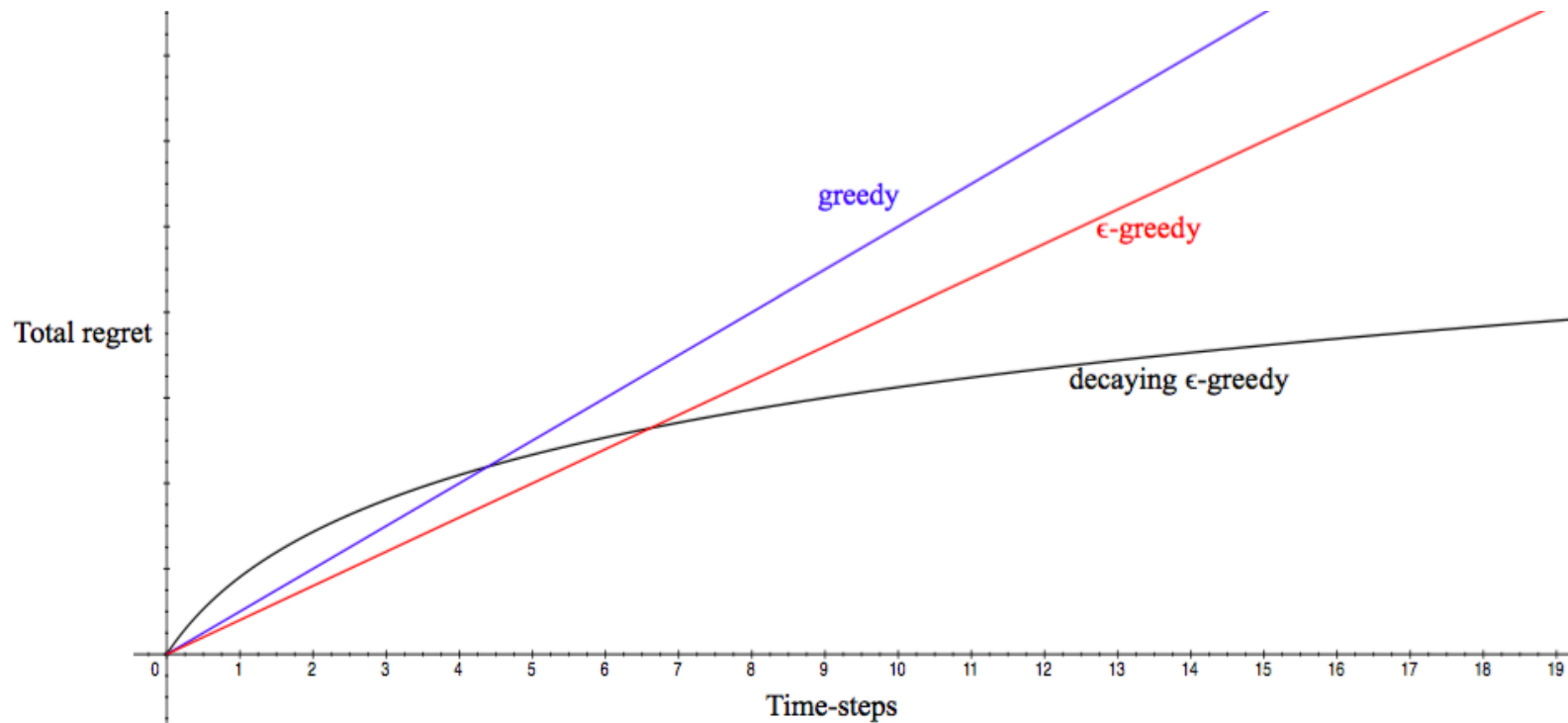
- ▶ The **count** $N_t(a)$: the number of times that action a has been selected prior to time t
- ▶ The **gap** Δ_a is the difference in value between action a and optimal action a^* : $\Delta_a = V^* - Q(a)$

- ▶ Regret is a function of gaps and the counts

$$\begin{aligned} L_t &= \mathbb{E} \left[\sum_{\tau=1}^t V^* - Q(a_\tau) \right] \\ &= \sum_{a \in \mathcal{A}} \mathbb{E} [N_t(a)] (V^* - Q(a)) \\ &= \sum_{a \in \mathcal{A}} \mathbb{E} [N_t(a)] \Delta_a \end{aligned}$$

- ▶ A good algorithm ensures **small counts for large gaps**
- ▶ **Problem**: rewards, and therefore gaps, are not known in advance!

Counting Regret



- ▶ If an algorithm forever explores uniformly it will have linear total regret
- ▶ If an algorithm never explores it will have linear total regret
- ▶ Is it possible to achieve sub-linear total regret?

Greedy Algorithm

- ▶ We consider algorithms that estimate: $\hat{Q}_t(a) \approx Q(a)$

- ▶ Estimate the value of each action by Monte-Carlo evaluation:

$$\hat{Q}_t(a) = \frac{1}{N_t(a)} \sum_{i=1}^t r_i \mathbf{1}(a_i = a) \quad \leftarrow \text{Sample average}$$

- ▶ The **greedy algorithm** selects action with highest estimated value

$$a_t^* = \operatorname{argmax}_{a \in \mathcal{A}} \hat{Q}_t(a)$$

- ▶ Greedy can lock onto a suboptimal action forever
- ▶ \Rightarrow Greedy has linear (in time) total regret

ϵ -Greedy Algorithm

- ▶ The ϵ -greedy algorithm continues to explore forever
 - With probability $(1 - \epsilon)$ select $a = \operatorname{argmax}_{a \in \mathcal{A}} \hat{Q}(a)$
 - With probability ϵ select a random action
- ▶ Constant ϵ ensures **expected regret at each time step** is:

$$l_t \geq \frac{\epsilon}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \Delta_a$$

- ▶ \Rightarrow ϵ -greedy has linear (in time) expected total regret

ε -Greedy Algorithm

A simple bandit algorithm

Initialize, for $a = 1$ to k :

$$Q(a) \leftarrow 0$$

$$N(a) \leftarrow 0$$

Repeat forever:

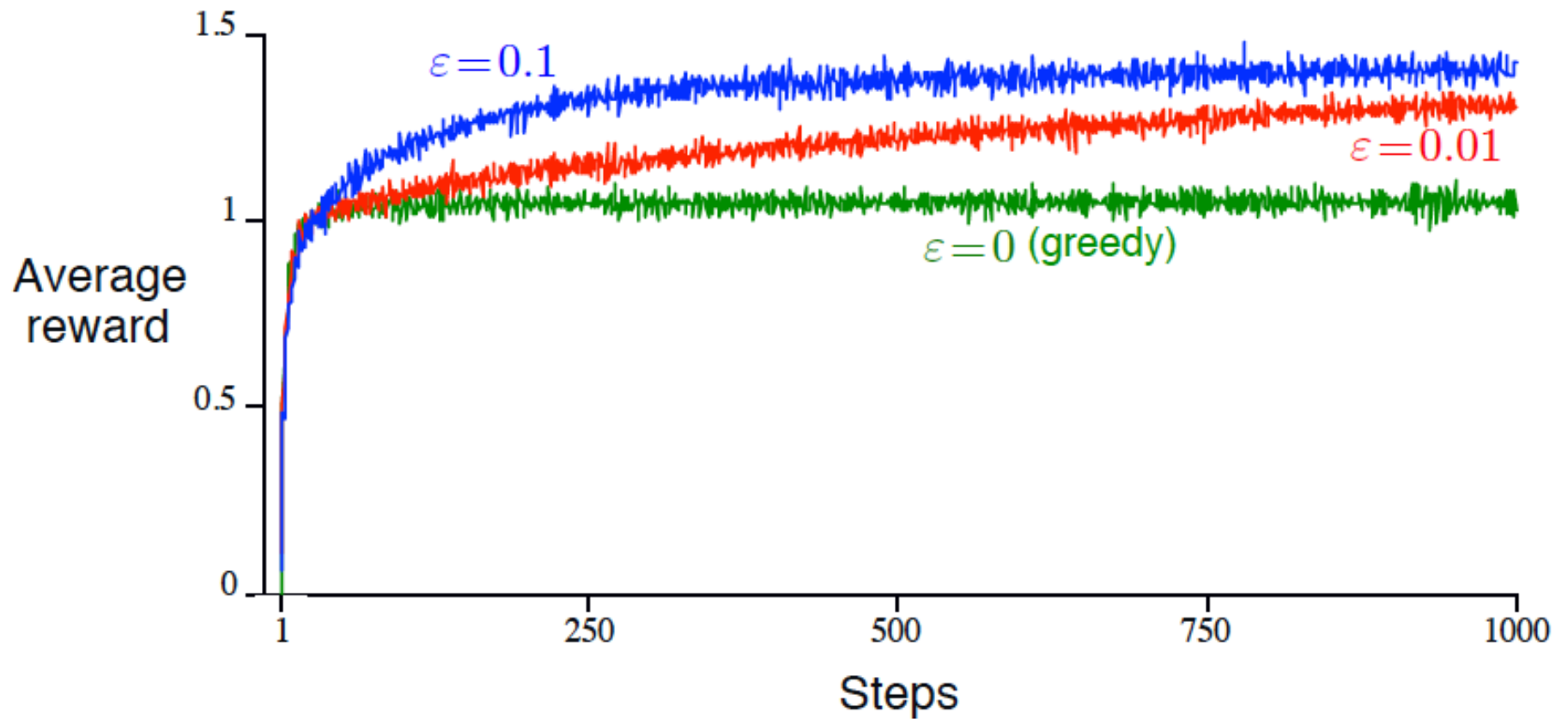
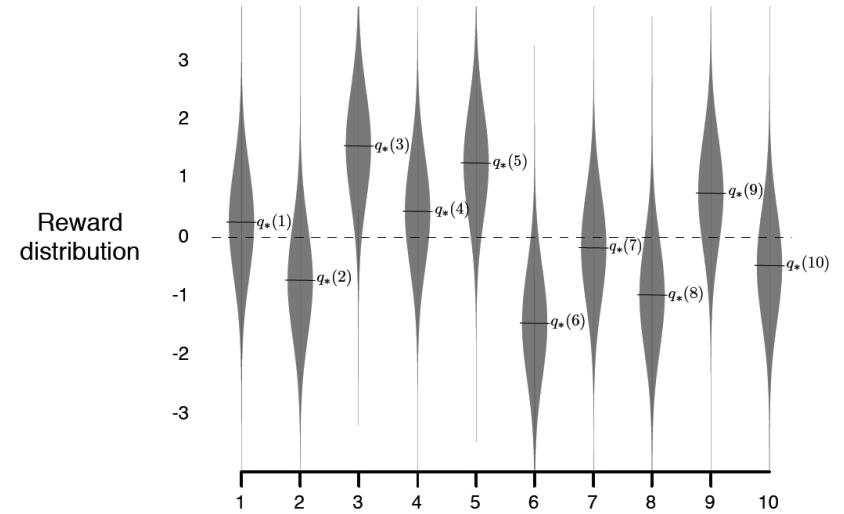
$$A \leftarrow \begin{cases} \arg \max_a Q(a) & \text{with probability } 1 - \varepsilon \quad (\text{breaking ties randomly}) \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$$R \leftarrow \text{bandit}(A)$$

$$N(A) \leftarrow N(A) + 1$$

$$Q(A) \leftarrow Q(A) + \frac{1}{N(A)} [R - Q(A)]$$

Average reward for three algorithms



Non-Stationary Worlds

- ▶ What if reward function changes over time?
- ▶ Then we should base reward estimates on more recent experience

▶ Starting with $Q_{n+1} = \frac{1}{n} \sum_{i=1}^n R_i$

just the incremental
calculation of sample mean

$$= Q_n + \frac{1}{n} [R_n - Q_n],$$

- ▶ We can up-weight influence of newer examples

$$Q_{n+1} = Q_n + \alpha [R_n - Q_n]$$

influence decays
exponentially in time!

$$= (1 - \alpha)^n Q_1 + \sum_{i=1}^n \alpha (1 - \alpha)^{n-i} R_i$$

Non-Stationary Worlds

- ▶ We can up-weight influence of newer examples

$$\begin{aligned} Q_{n+1} &= Q_n + \alpha [R_n - Q_n] \\ &= (1 - \alpha)^n Q_1 + \sum_{i=1}^n \alpha (1 - \alpha)^{n-i} R_i \end{aligned}$$

influence decays exponentially in time!

- ▶ Can even make α vary with step n and action a
- ▶ And still assure convergence so long as

$$\sum_{n=1}^{\infty} \alpha_n(a) = \infty \quad \text{and} \quad \sum_{n=1}^{\infty} \alpha_n^2(a) < \infty$$

big enough to overcome
initialization and random
fluctuations

small enough to eventually
converge

ε -Greedy Algorithm

A simple bandit algorithm

Initialize, for $a = 1$ to k :

$$Q(a) \leftarrow 0$$

$$N(a) \leftarrow 0$$

Repeat forever:

$$A \leftarrow \begin{cases} \arg \max_a Q(a) & \text{with probability } 1 - \varepsilon \quad (\text{breaking ties randomly}) \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$$R \leftarrow \text{bandit}(A)$$

$$N(A) \leftarrow N(A) + 1$$

$$Q(A) \leftarrow Q(A) + \frac{1}{N(A)} [R - Q(A)]$$

$$\alpha_n(a)$$

Back to stationary worlds ...

Optimistic Initialization

- ▶ **Simple and practical idea:** initialize $Q(a)$ to high value
- ▶ Update action value by incremental Monte-Carlo evaluation
- ▶ Starting with $N(a) > 0$

$$\hat{Q}_t(a_t) = \hat{Q}_{t-1} + \frac{1}{N_t(a_t)}(r_t - \hat{Q}_{t-1})$$

- ▶ Encourages systematic exploration early on
- ▶ But optimistic greedy can still lock onto a suboptimal action if rewards are stochastic

just an incremental estimate of sample mean, including one 'hallucinated' initial optimistic value

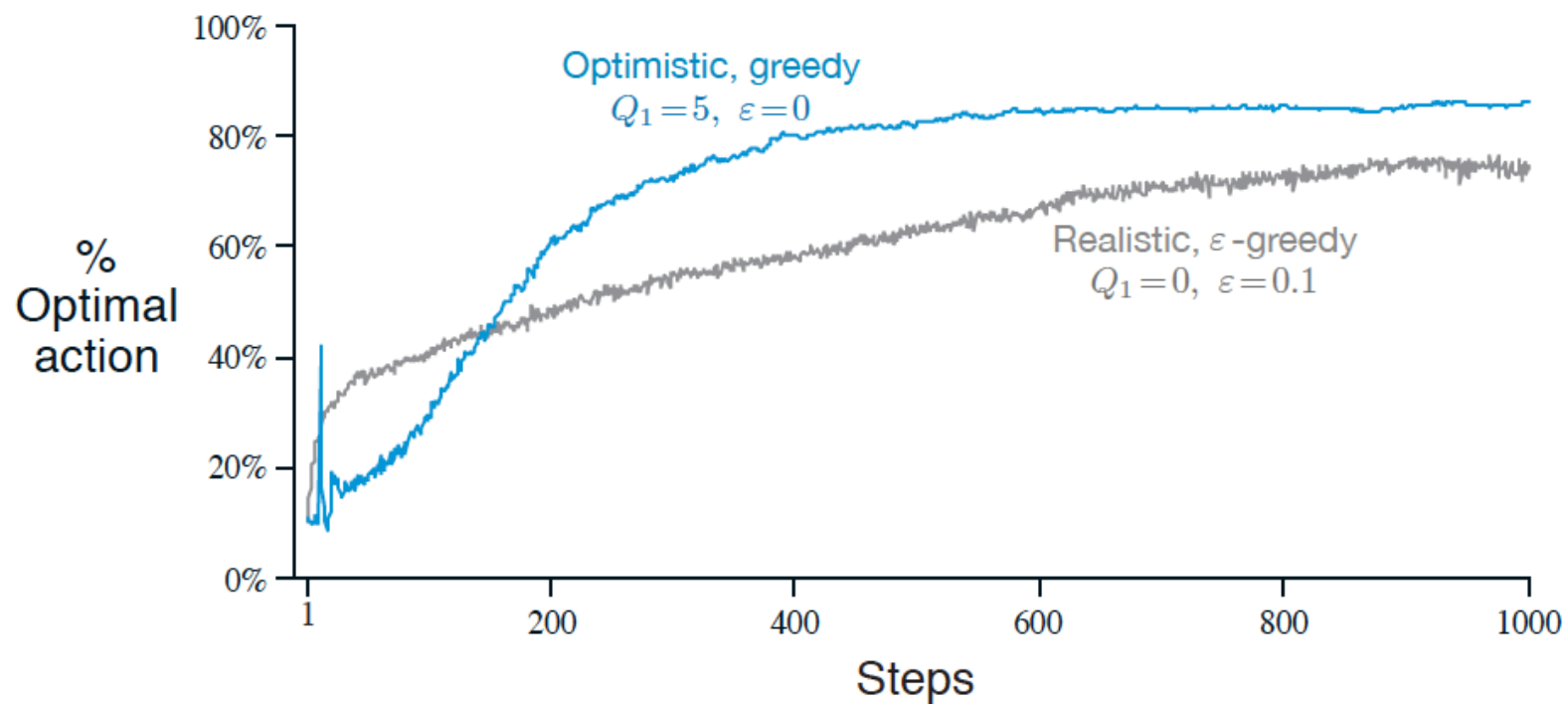


Figure 2.3: The effect of optimistic initial action-value estimates on the 10-armed testbed. Both methods used a constant step-size parameter, $\alpha = 0.1$.

Decaying ϵ_t -Greedy Algorithm

- ▶ Pick a decay schedule for $\epsilon_1, \epsilon_2, \dots$
- ▶ Consider the following schedule

$$c > 0$$


$$d = \min_{a|\Delta_a>0} \Delta_i$$

$$\epsilon_t = \min \left\{ 1, \frac{c|\mathcal{A}|}{d^2 t} \right\}$$

Smallest non-zero gap



How does ϵ change as smallest non-zero gap shrinks?



- ▶ Decaying ϵ_t -greedy has **logarithmic asymptotic total regret**
- ▶ Unfortunately, schedule requires advance knowledge of gaps
- ▶ **Goal:** find an algorithm with sub-linear regret for any multi-armed bandit (without knowledge of R)

Upper Confidence Bounds

- ▶ Estimate an **upper confidence** $U_t(a)$ for each action value
- ▶ Such that with high probability

$$Q(a) \leq \hat{Q}_t(a) + \hat{U}_t(a)$$

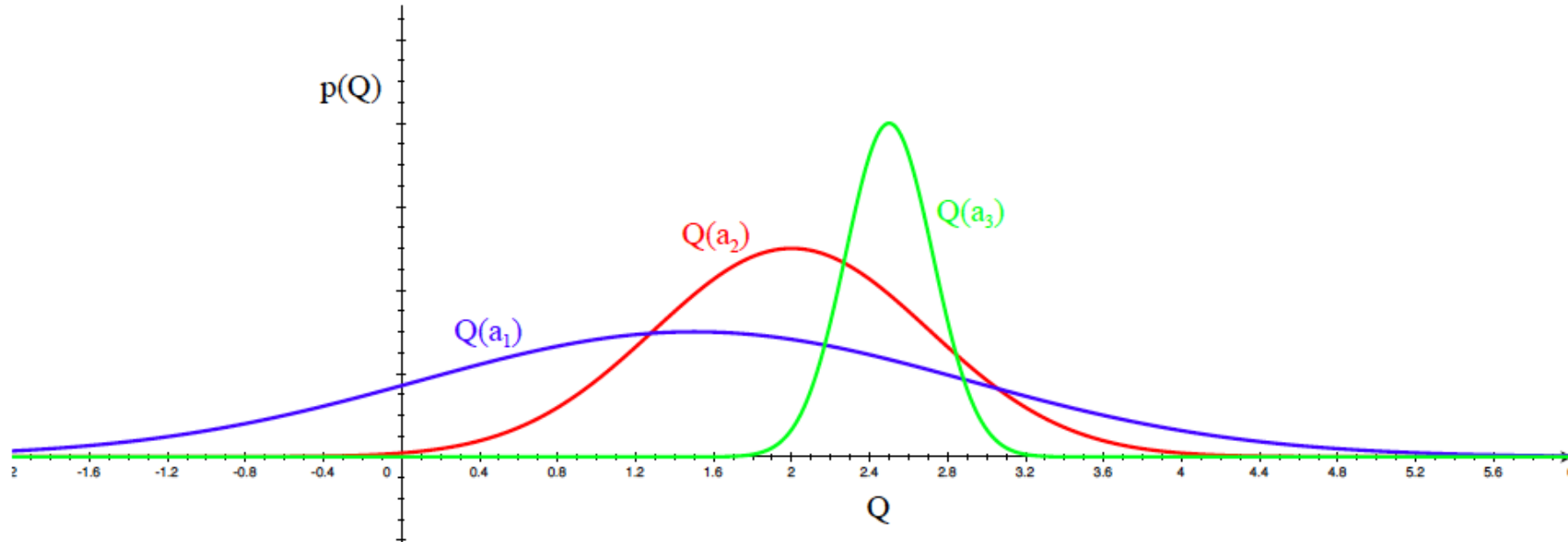
Estimated mean

Estimated Upper
Confidence interval

- ▶ This depends on the number of times $N(a)$ has been selected
 - Small $N_t(a) \Rightarrow$ large $U_t(a)$ (estimated value is uncertain)
 - Large $N_t(a) \Rightarrow$ small $U_t(a)$ (estimated value is more accurate)
- ▶ Select action maximizing **Upper Confidence Bound** (UCB)

$$a_t = \operatorname{argmax}_{a \in \mathcal{A}} \hat{Q}_t(a) + \hat{U}_t(a)$$

Optimism in the Face of Uncertainty



- ▶ This depends on the number of times $N(a_k)$ has been selected
 - Small $N_t(a_k) \Rightarrow$ upper bound will be far from sample mean
 - Large $N_t(a_k) \Rightarrow$ upper bound will be closer to sample mean

but how can we calculate upper bound if we don't know form of $P(Q)$?

Hoeffding's Inequality

Theorem (Hoeffding's Inequality)

Let X_1, \dots, X_t be i.i.d. random variables in $[0,1]$, and let $\bar{X}_t = \frac{1}{t} \sum_{\tau=1}^t X_\tau$ be the sample mean. Then

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

- ▶ We will apply Hoeffding's Inequality to rewards of the bandit conditioned on selecting action a

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

Calculating Upper Confidence Bounds

- ▶ Pick a probability p that true value exceeds UCB
- ▶ Now solve for $U_t(a)$

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

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

- ▶ **Reduce p** as we observe more rewards, e.g. $p = t^{-c}$, $c=4$
(note: c is a hyper-parameter that trades-off explore/exploit)
- ▶ Ensures we select optimal action as $t \rightarrow \infty$

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

UCB1 Algorithm

- ▶ This leads to the **UCB1 algorithm**

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

Theorem

The UCB algorithm achieves logarithmic asymptotic total regret

$$\lim_{t \rightarrow \infty} L_t \leq 8 \log t \sum_{a | \Delta_a > 0} \Delta_a$$

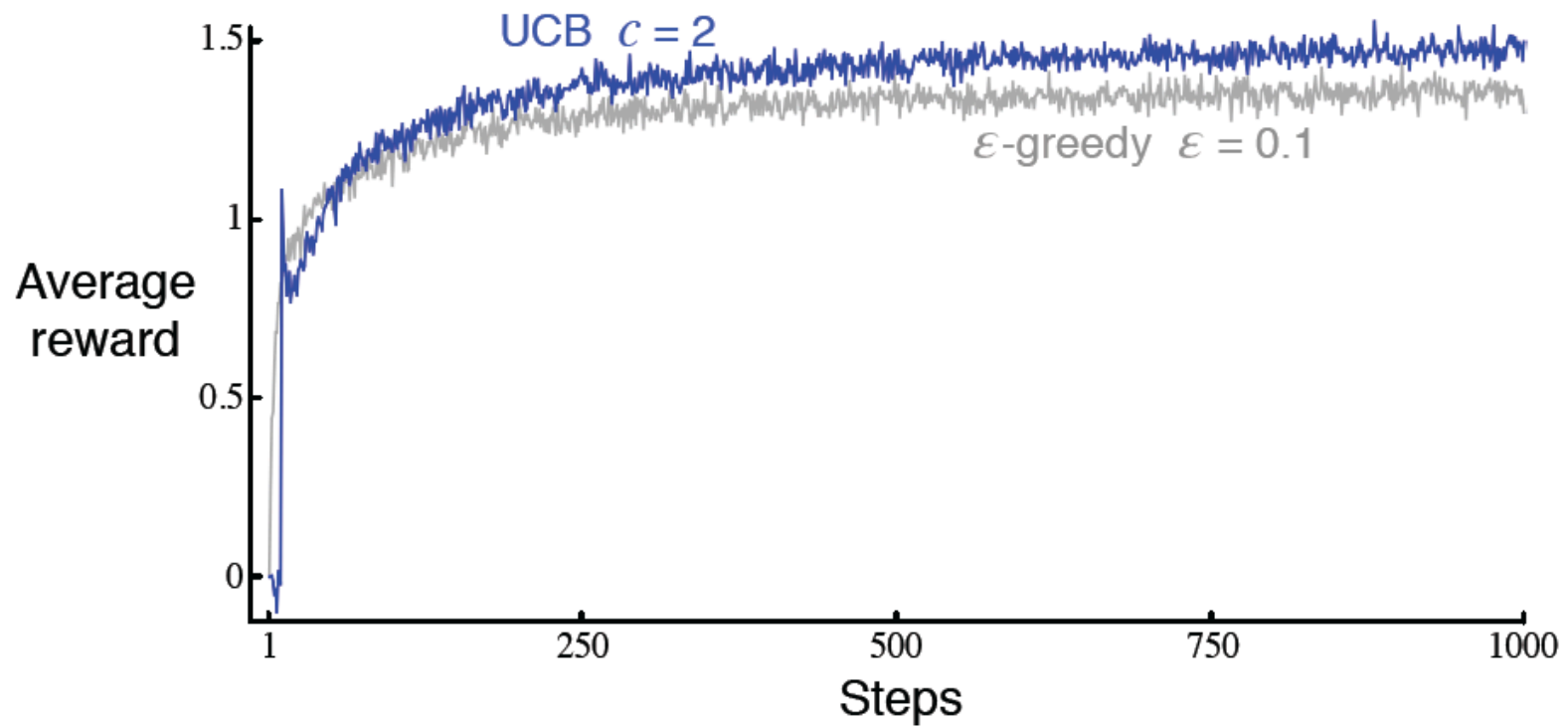


Figure 2.4: Average performance of UCB action selection on the 10-armed testbed. As shown, UCB generally performs better than ϵ -greedy action selection, except in the first k steps, when it selects randomly among the as-yet-untried actions.

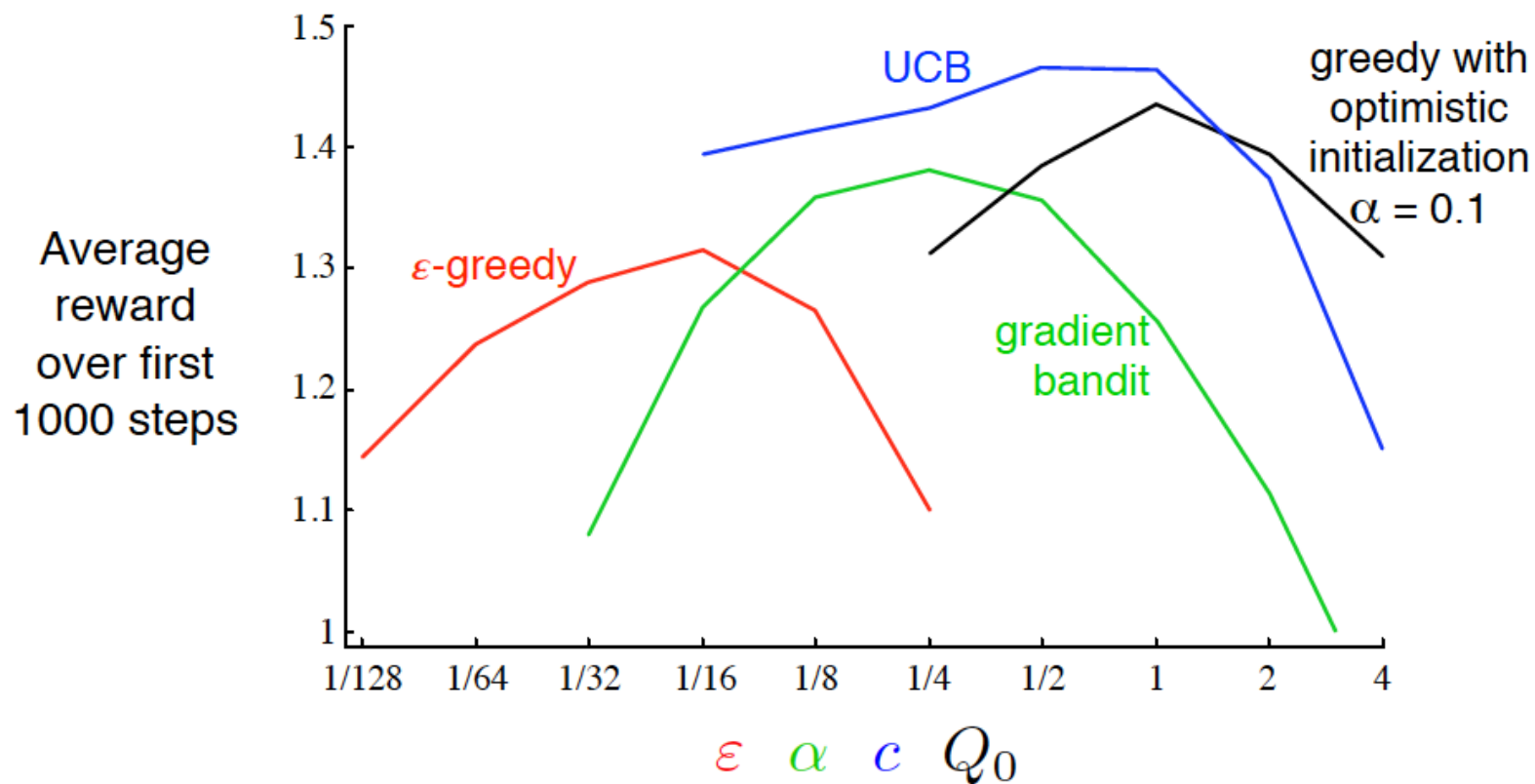


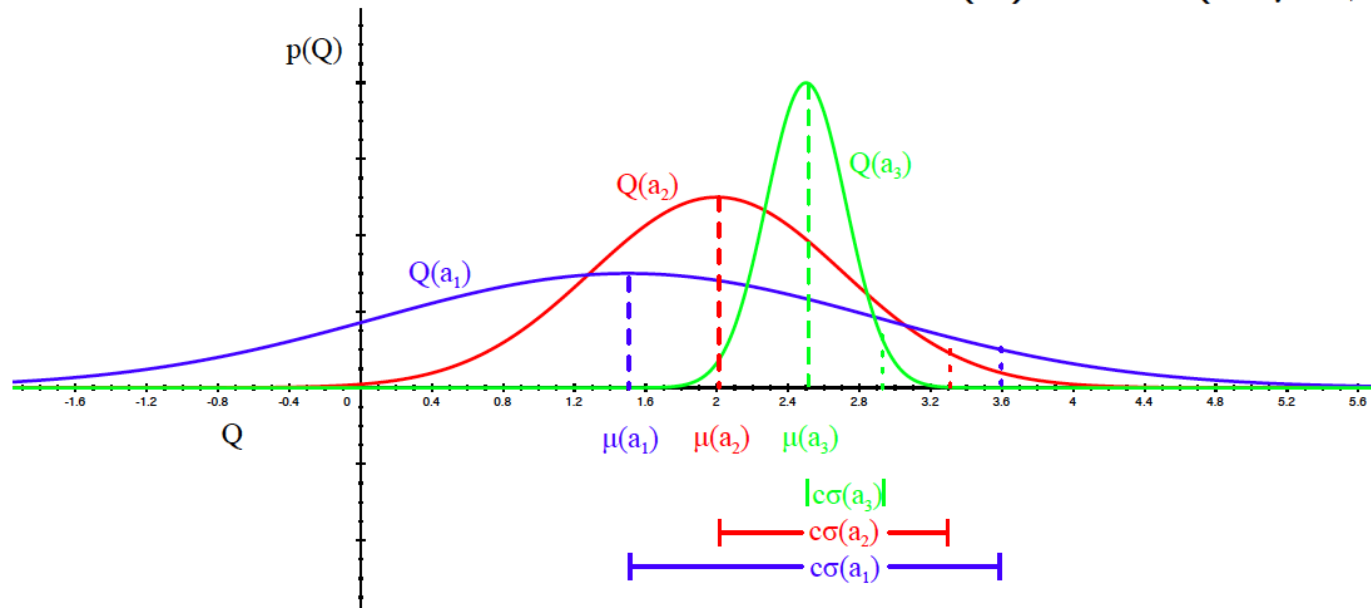
Figure 2.6: A parameter study of the various bandit algorithms presented in this chapter. Each point is the average reward obtained over 1000 steps with a particular algorithm at a particular setting of its parameter.

Bayesian Bandits

- ▶ So far we have made no assumptions about the reward distribution R
 - Except bounds on rewards
- ▶ Bayesian bandits exploit prior knowledge of rewards, $p[\mathcal{R}]$
- ▶ They compute posterior distribution of rewards $p[\mathcal{R} | h_t]$
 - where the history is: $h_t = a_1, r_1, \dots, a_{t-1}, r_{t-1}$
- ▶ Use posterior to guide exploration
 - Upper confidence bounds (Bayesian UCB)
 - Can avoid weaker, assumption free, Hoeffding bounds
- ▶ Better performance if prior knowledge is accurate

Bayesian UCB Example

- Assume reward distribution is Gaussian, $\mathcal{R}_a(r) = \mathcal{N}(r; \mu_a, \sigma_a^2)$



- Compute Gaussian posterior over μ_a and σ_a^2 (by Bayes law)

$$p[\mu_a, \sigma_a^2 | h_t] \propto p[\mu_a, \sigma_a^2] \prod_{t | a_t=a} \mathcal{N}(r_t; \mu_a, \sigma_a^2)$$

- Pick action $a_t = \operatorname{argmax}_a \mu_a + c\sigma_a / \sqrt{N(a)}$

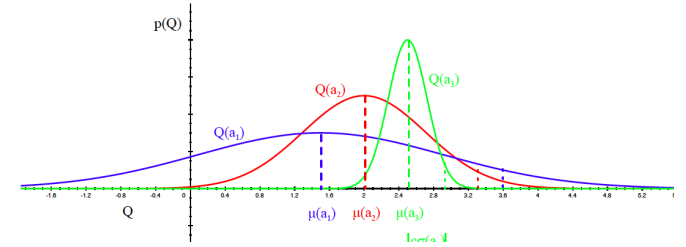
Probability Matching

- ▶ **Probability matching** selects action a according to probability that a is the optimal action

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

- ▶ Probability matching is naturally optimistic in the face of uncertainty
 - Uncertain actions have higher probability of being max
- ▶ Can be difficult to compute analytically.

Thompson Sampling



- ▶ Thompson sampling implements probability matching

$$\begin{aligned}\pi(a \mid h_t) &= \mathbb{P} [Q(a) > Q(a'), \forall a' \neq a \mid h_t] \\ &= \mathbb{E}_{\mathcal{R} \mid h_t} \left[\mathbf{1}(a = \operatorname{argmax}_{a \in \mathcal{A}} Q(a)) \right]\end{aligned}$$

- ▶ here \mathcal{R} is the actual (unknown) distribution from which rewards are drawn
- ▶ Use Bayes law to compute **posterior distribution** : $p[\mathcal{R} \mid h_t]$ (i.e., distribution over the parameters of \mathcal{R})
- ▶ **Sample** a reward distribution R from posterior
- ▶ Compute action-value function: $Q(a) = \mathbb{E}[\mathcal{R}_a]$
- ▶ Select action maximizing value on sample: $a_t = \operatorname{argmax}_{a \in \mathcal{A}} Q(a)$

Contextual Bandits (aka Associative Search)

- ▶ A contextual bandit is a tuple $\langle \mathcal{A}, \mathcal{S}, \mathcal{R} \rangle$
- ▶ \mathcal{A} is a known set of k actions (or “arms”)
- ▶ $\mathcal{S} = \mathbb{P}[s]$ is an unknown distribution over states (or “contexts”)
- ▶ $\mathcal{R}_s^a(r) = \mathbb{P}[r|s, a]$ is an unknown probability distribution over rewards
- ▶ At each time t
 - Environment generates state $s_t \sim \mathcal{S}$
 - Agent selects action $a_t \in \mathcal{A}$
 - Environment generates reward $r_t \sim \mathcal{R}_{s_t}^{a_t}$
- ▶ **The goal** is to maximize cumulative reward $\sum_{\tau=1}^t r_\tau$



Value of Information

- ▶ Exploration is useful because it gains information
- ▶ Can we quantify the value of information?
 - How much reward a decision-maker would be prepared to pay in order to have that information, prior to making a decision
 - **Long-term reward** after getting information vs. **immediate reward**
- ▶ Information gain is higher in uncertain situations
- ▶ Therefore it makes sense to explore uncertain situations more
- ▶ If we know value of information, we can trade-off exploration and exploitation optimally

Information State Search in MDPs

- ▶ MDPs can be augmented to include information state
- ▶ Now the augmented state is $\tilde{S} = \langle s, s^\sim \rangle$
 - where s is **original state** within MDP
 - and s^\sim is a **statistic of the history** (accumulated information)
- ▶ Each action a causes a transition
 - to a new state s' with probability $\mathcal{P}_{s,s'}^a$
 - to a new information state s'^\sim $\tilde{\mathcal{P}}_{\tilde{s},\tilde{s}'}^a$
- ▶ Defines MDP in augmented information state space

$$\tilde{\mathcal{M}} = \langle \tilde{S}, \mathcal{A}, \tilde{\mathcal{P}}, \mathcal{R}, \gamma \rangle$$

Conclusion

- ▶ Have covered several principles for exploration/exploitation
 - Naive methods such as ϵ -greedy
 - Optimistic initialization
 - Upper confidence bounds
 - Probability matching
 - Information State Search
- ▶ These principles were developed in bandit setting
- ▶ But same principles also apply to MDP setting

Thank you