10703 Deep Reinforcement Learning and Control

Russ Salakhutdinov

Machine Learning Department
rsalakhu@cs.cmu.edu

Exploration and Exploitation

Used Materials

- **Disclaimer**: Much of the material and slides for this lecture were borrowed from Rich Sutton’s class and David Silver’s class on Reinforcement Learning.

Exploration vs. Exploitation Dilemma

- Online decision-making involves a fundamental choice:
  - Exploitation: Make the best decision given current information
  - Exploration: Gather more information

- The best long-term strategy may involve short-term sacrifices
- Gather enough information to make the best overall 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 ⟨A, R⟩
- 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
- 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 $\sum_{\tau=1}^{t} r_\tau$

Regret

- The action-value is the mean reward for action a,

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

- The optimal value V* is

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

- The regret is the opportunity loss for one step

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

- The total regret is the total opportunity loss

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

- Maximize cumulative reward = minimize total regret

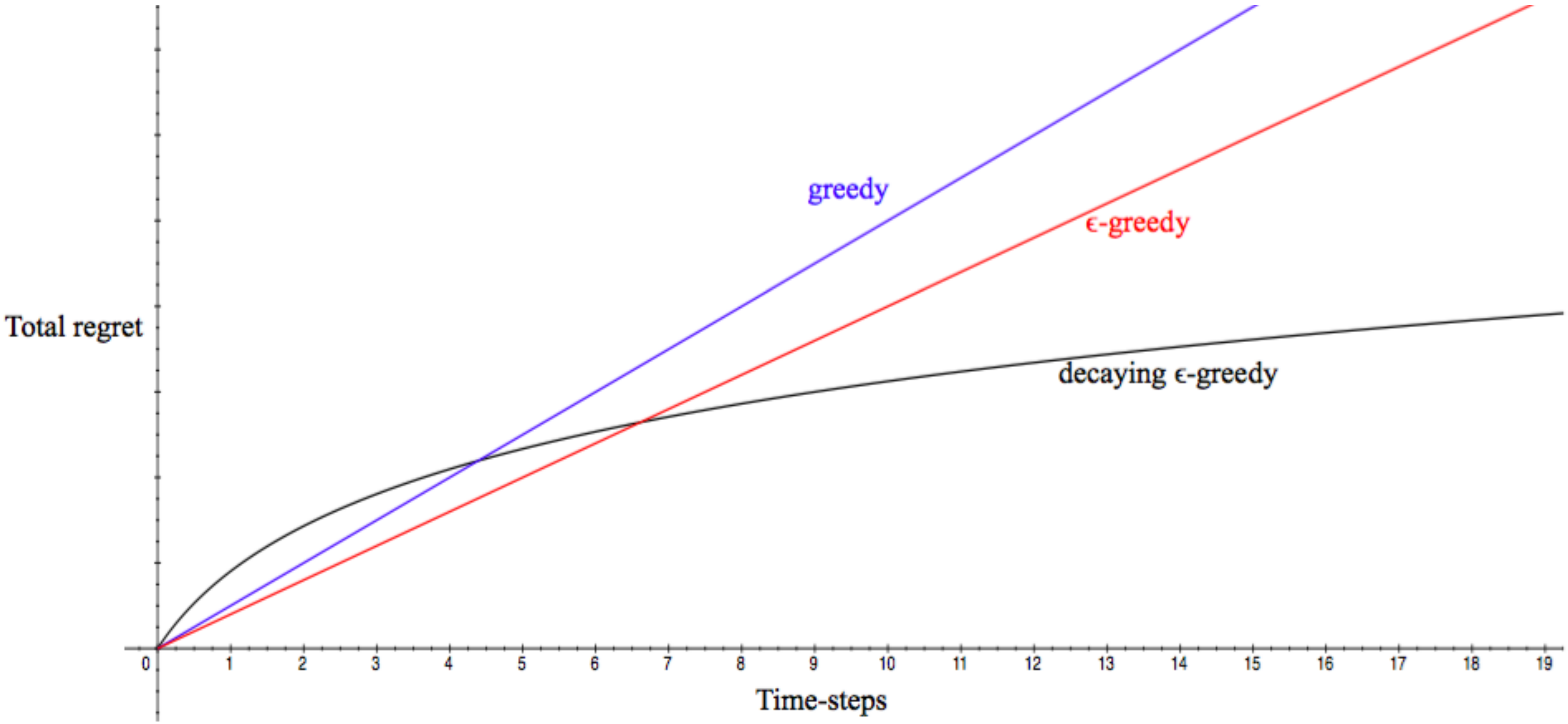
Counting Regret

- 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

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

- A good algorithm ensures small counts for large gaps
- Problem: gaps are not known!

Counting Regret



- If an algorithm forever explores 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:

$$\hat{Q}_t(a) = \frac{1}{N_t(s)} \sum_{i=1}^{t} r_i \mathbf{1}(a_i = a)$$

Sample average

- The greedy algorithm selects action with highest value

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

- Greedy can lock onto a suboptimal action forever
- ⇒ Greedy has linear total regret

ε-Greedy Algorithm

- The ε-greedy algorithm continues to explore forever
  - With probability 1 − ε select $a = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \hat{Q}(a)$
  - With probability ε select a random action
- Constant ε ensures minimum regret

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

- ⇒ ε-greedy has linear 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:

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

$R \leftarrow bandit(A)$

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

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

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 can still lock onto suboptimal action

Decaying ε_t-Greedy Algorithm

- Pick a decay schedule for ε_1, ε_2, ...
- Consider the following schedule

Smallest non-zero gap

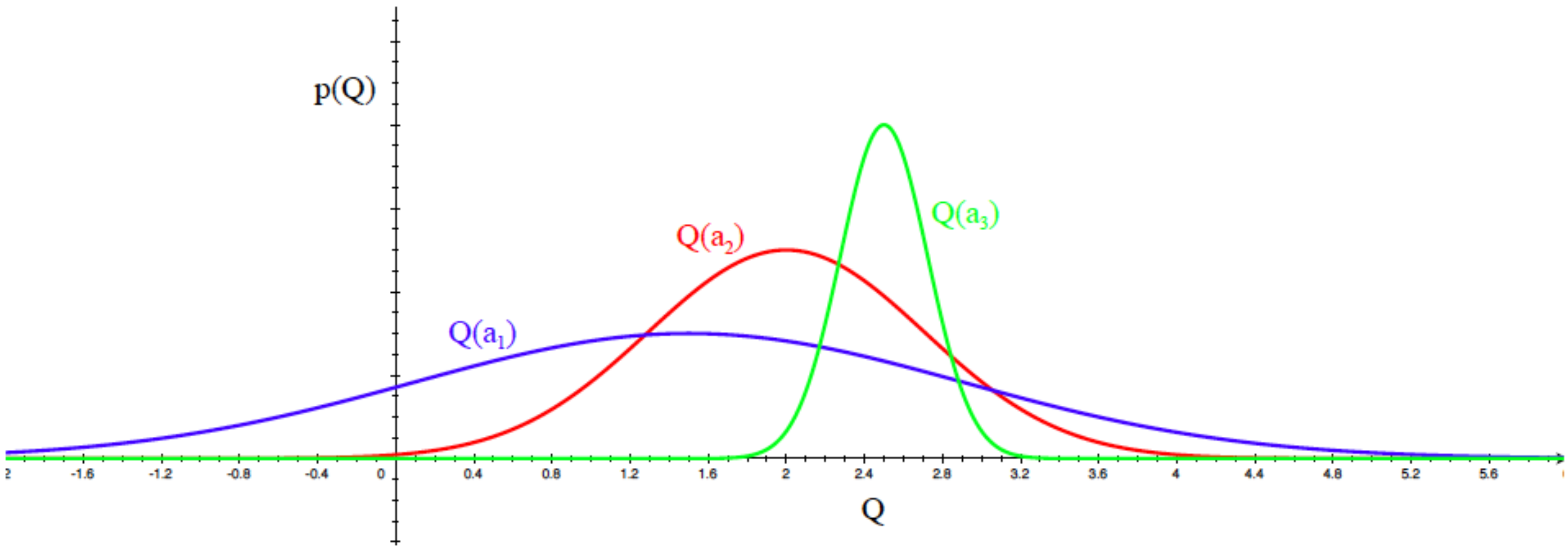
$$c > 0$$

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

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

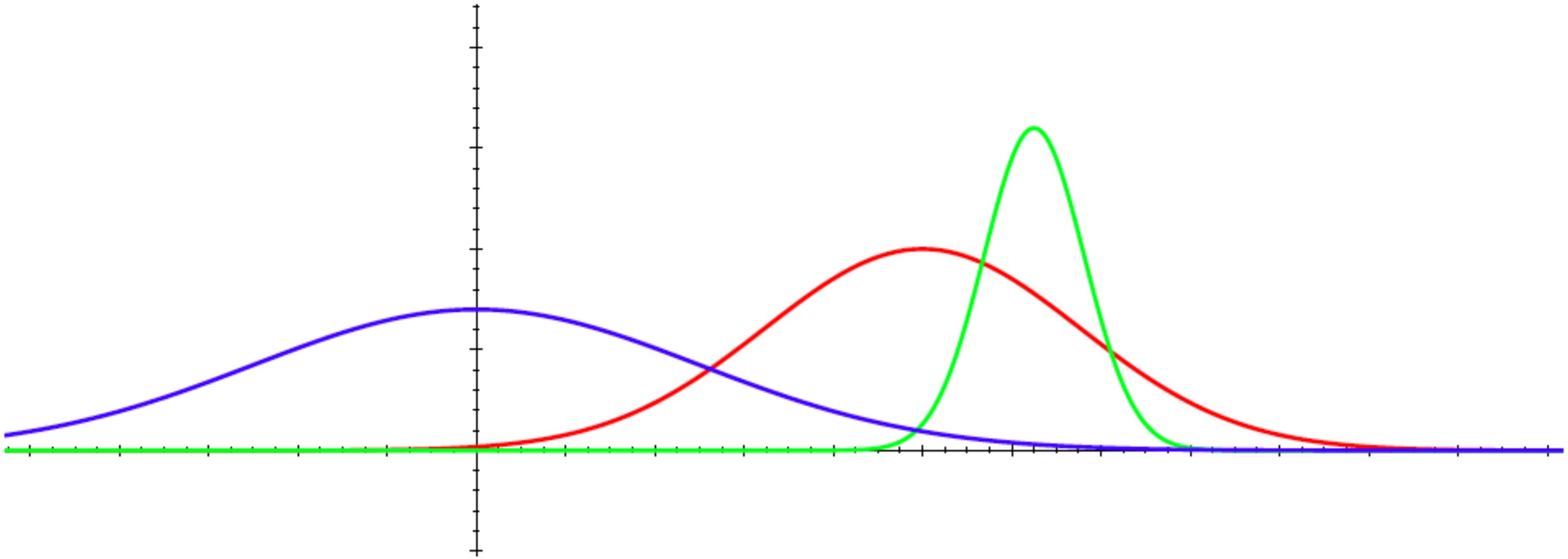
- Decaying ε_t-greedy has logarithmic asymptotic total regret
- Unfortunately, schedule requires advance knowledge of gaps
- Goal: find an algorithm with sub-linear regret for any multi-armed bandit (without knowledge of R)

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 explore 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
- Until we home in on best action

Upper Confidence Bounds

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

$$Q(a) \leq \underbrace{\hat{Q}_t(a)}_{\text{Estimated mean}} + \underbrace{\hat{U}_t(a)}_{\text{Estimated Upper Confidence}}$$

- 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 accurate)
- Select action maximizing Upper Confidence Bound (UCB)

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

Hoeffding's Inequality

Theorem (Hoeffding's Inequality)

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

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

- 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 $t \to \infty$

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

UCB1 Algorithm

- This leads to the UCB1 algorithm

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

Theorem

The UCB algorithm achieves logarithmic asymptotic total regret

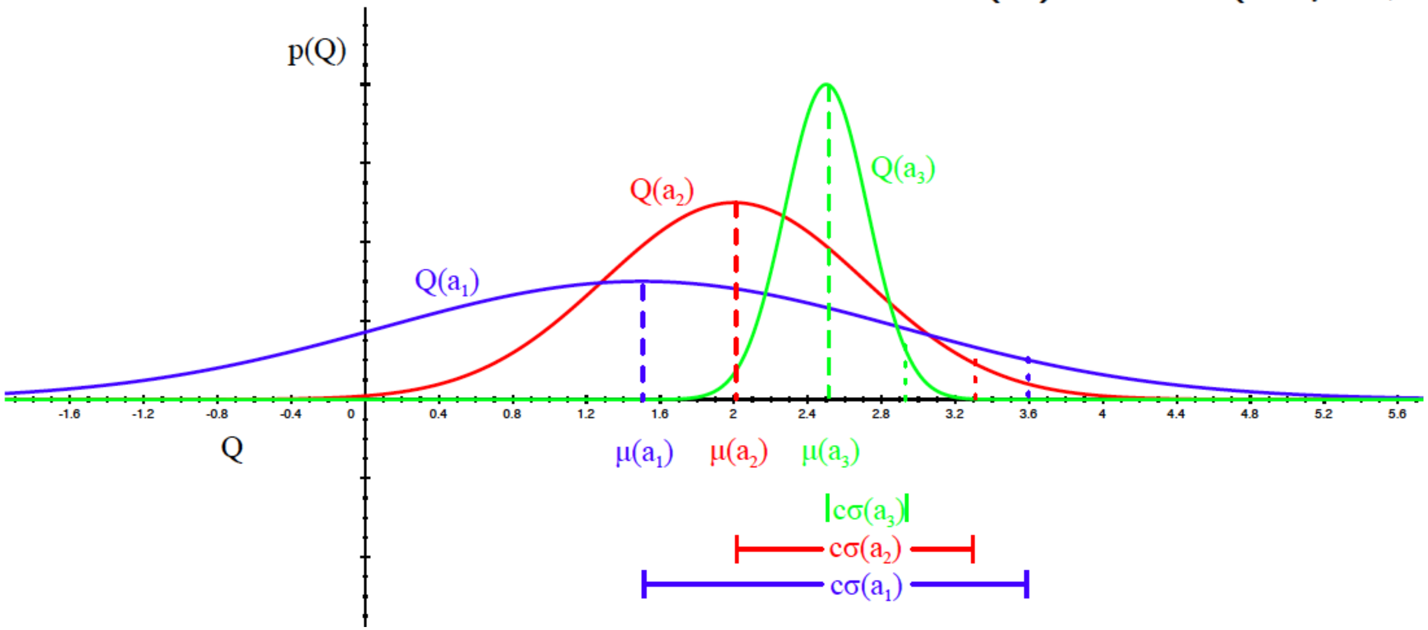
$$\lim_{t \to \infty} L_t \leq 8 \log t \sum_{a | \Delta_a > 0} \Delta_a$$

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 $p[\mathcal{R} \mid h_t]$
  - where the history is: $h_t = a_1, r_1, ..., a_{t-1}, r_{t-1}$
- Use posterior to guide exploration
  - Upper confidence bounds (Bayesian UCB)
  - Probability matching (Thompson sampling)
- 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)

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

- Pick action that maximizes standard deviation of Q(a)

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

Probability Matching

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

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

- Probability matching is 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

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

- Use Bayes law to compute posterior distribution: $p\left[\mathcal{R} \mid h_t\right]$
- Sample a reward distribution R from posterior
- Compute action-value function: $Q(a) = \mathbb{E}\left[\mathcal{R}_a\right]$
- Select action maximizing value on sample: $a_t = \operatorname*{argmax}_{a \in \mathcal{A}} Q(a)$

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

Contextual Bandits

- A contextual bandit is a tuple ⟨A, S , R⟩
- 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 $\sum_{\tau=1}^{t} r_\tau$

Exploration/Exploitation for MDPs

- The same principles for exploration/exploitation apply to MDPs
  - Naive Exploration
  - Optimistic Initialization
  - Optimism in the Face of Uncertainty
  - Probability Matching
  - Information State Search

Optimistic Initialization: Model-Free RL

- Initialize action-value function Q(s,a) to $\frac{r_{max}}{1-\gamma}$
- Run favorite model-free RL algorithm
  - Monte-Carlo control
  - Sarsa
  - Q-learning
  - …
- Encourages systematic exploration of states and actions

Optimistic Initialization: Model-Based RL

- Construct an optimistic model of the MDP
- Initialize transitions to go to heaven
  - (i.e. transition to terminal state with r_{max} reward)
- Solve optimistic MDP by favourite planning algorithm
  - policy iteration
  - value iteration
  - tree search
  - …
- Encourages systematic exploration of states and actions
- e.g. RMax algorithm (Brafman and Tennenholtz)

Upper Confidence Bounds: Model-Free RL

- Maximize UCB on action-value function $Q^{\pi}(s,a)$

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\; Q(s_t, a) + U(s_t, a)$$

- Remember UCB1 Algorithm:

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

How do we estimate the counts in continuous spaces?

Bayesian Model-Based RL

- Maintain posterior distribution over MDP models
- Estimate both transitions and rewards, $p[\mathcal{P}, \mathcal{R} \mid h_t]$
  - where the history is: $h_t = a_1, r_1, ..., a_{t-1}, r_{t-1}$
- Use posterior to guide exploration
  - Upper confidence bounds (Bayesian UCB)
  - Probability matching (Thompson sampling)

Thompson Sampling: Model-Based RL

- Thompson sampling implements probability matching

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

- Use Bayes law to compute posterior distribution: $p\left[\mathcal{P}, \mathcal{R} \mid h_t\right]$
- Sample from posterior an MDP P, R
- Solve MDP using favorite planning algorithm to get $Q^*(s, a)$
- Select optimal action for sampled MDP,

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ Q^*(s_t, a)$$

Information State Search in MDPs

- MDPs can be augmented to include information state
- Now the augmented state is ⟨s,s~⟩
  - where s is original state within MDP
  - and s~ is a statistic of the history (accumulated information)
- Each action a causes a transition
  - to a new state s′ with probability $\mathcal{P}^a_{s,s'}$
  - to a new information state s~′ $\tilde{\mathcal{P}}^a_{\tilde{s},\tilde{s}'}$
- Defines MDP in augmented information state space

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

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