Lecture 16: Bandits

CS 486/686: Introduction to Artificial Intelligence

Plan for Today

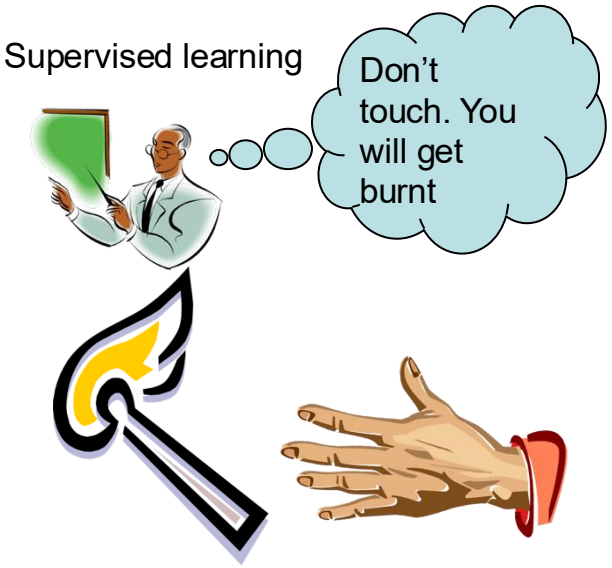
- What is Reinforcement Learning
- Multi-armed Bandits

What is RL?

- Reinforcement learning is learning what to do so as to maximize a numerical reward signal
- Learner is no told what actions to take
- Learner discovers value of actions by
  - Trying actions out
  - Seeing what the reward is

What is RL?

Supervised learning

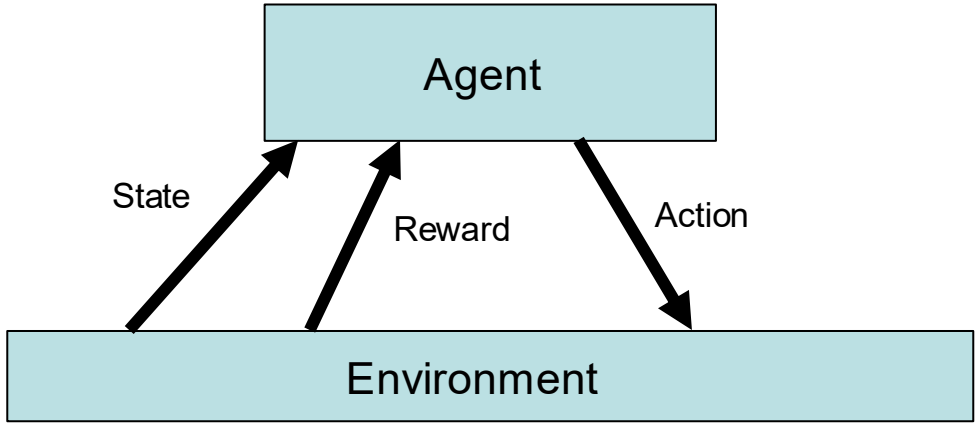


Reinforcement learning



Ouch!

Reinforcement Learning Problem



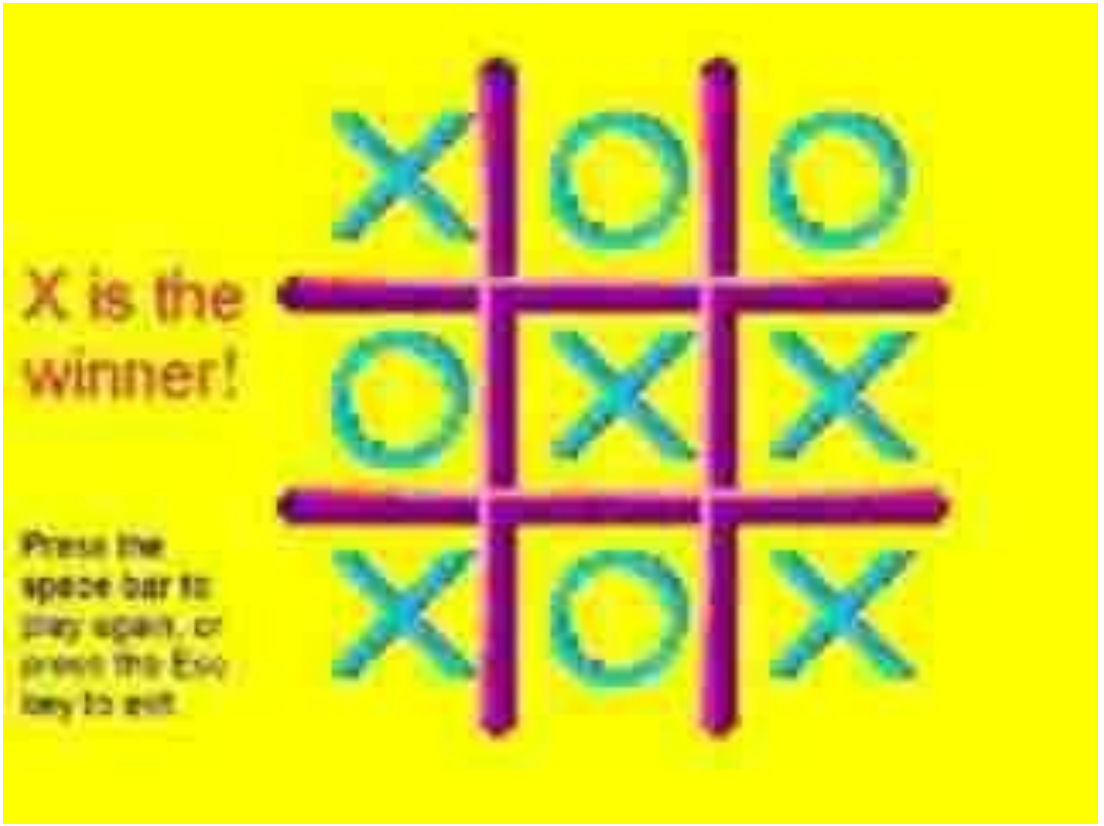
Goal: Learn to choose actions that maximize $r_0+\gamma\, r_1+\gamma^2 r_2+\ldots$, where $0\cdot<\gamma<1$

RL Model

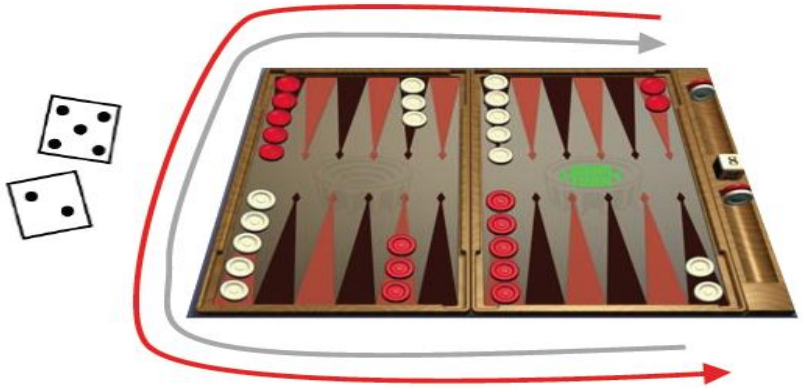
- Set of states S
- Set of actions A
- Set of reinforcement signals (rewards)
  - Rewards may be delayed

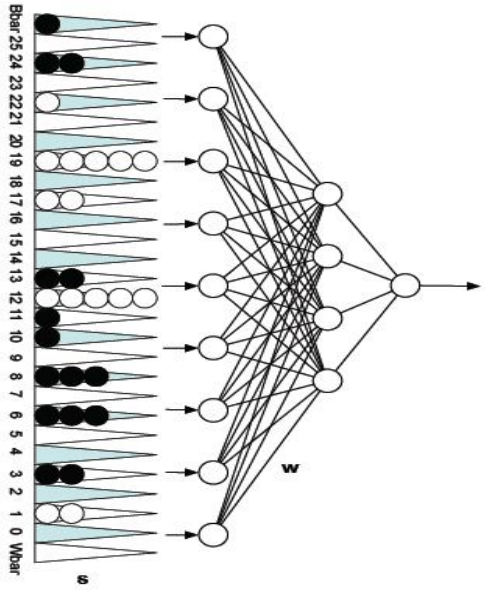
Example: Tic-Tac-Toe

- State: Board configuration
- Actions: Next move
- Rewards: 1 for a win, -1 for a loss, 0 for a draw
- Problem: find π: $S \mapsto A$ such that it maximizes reward



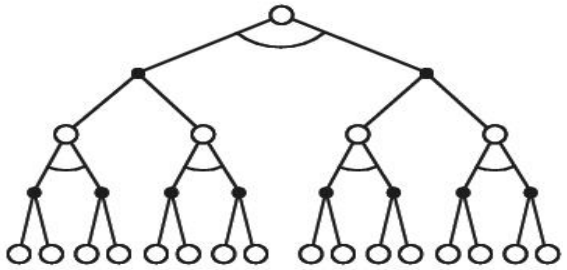
Example: TD-Gammon





estimated state value
(≈ prob of winning)

Action selection
by a shallow search



Start with a random Network

Play millions of games against itself

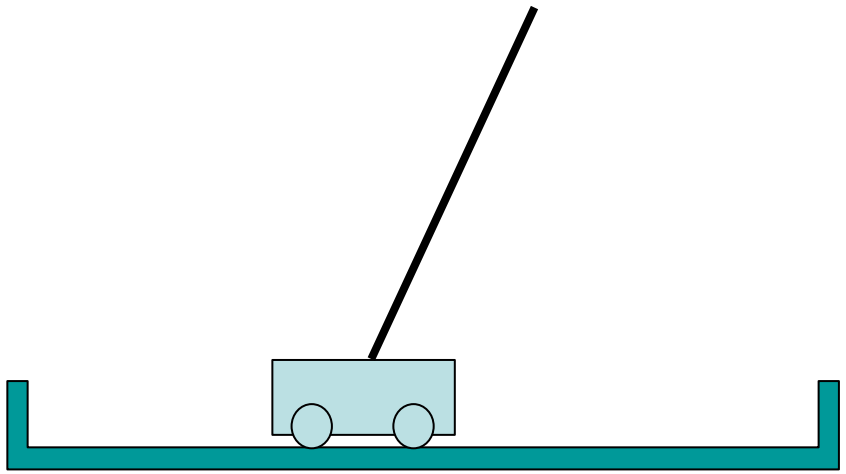
Learn a value function from this simulated experience

Six weeks later it's the best player of backgammon in the world

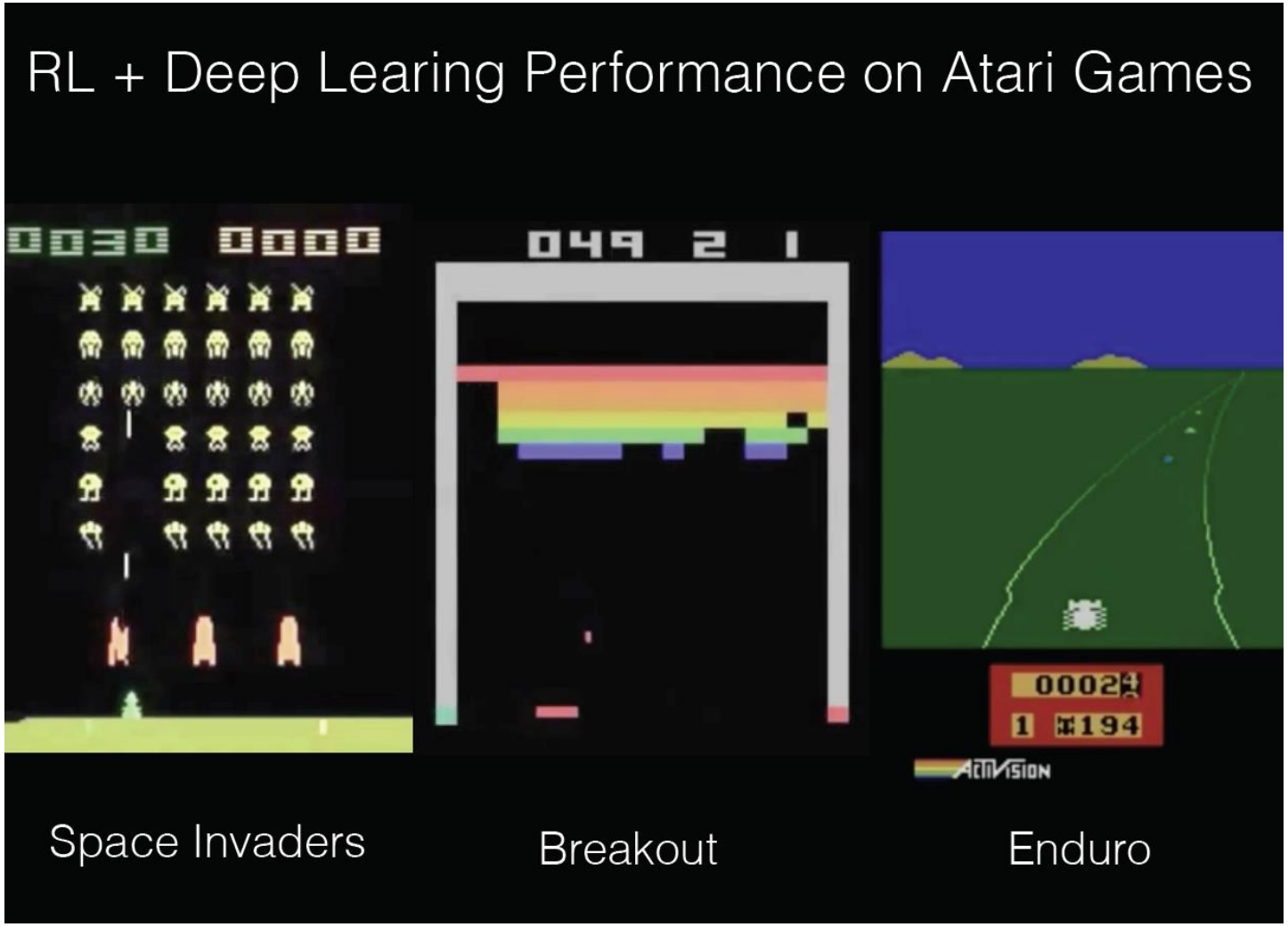
Originally used expert handcrafted features, later repeated with raw board positions

Example: Inverted Pendulum

- State: $x(t), x'(t), \theta(t), \theta'(t)$
- Actions: Force F
- Reward: +1 for each step the pole is balanced
- Problem: Find π: $S \mapsto A$ that maximizes reward



Example: Atari Games



Other Applications

- Robotics
  - Dexterous manipulations, drone maneuvers
- Datacenter management
  - Policies for bringing up and shutting down machines
  - Center cooling
- Video compression
- Discovering new algorithms
- Routing Autonomous Vehicles
- Natural Language Processing (Reinforcement Learning with Human Feedback (RLHF))
- Finance (e.g. portfolio optimization, market making, option pricing)

RL Characteristics

- Delayed reward
  - Credit assignment problem
- Exploration and exploitation
- Possibility that a state is only partially observable
- Life-long learning

Multi-Armed Bandits

- Multi-Armed Bandits (MAB) are the simplest RL problem
- Widely used:
  - Experiment design (clinical trials)
  - Online ad placement
  - Recommender Systems
  - …



K-Armed Bandit

At each time step t, you select an action A_t from k possible actions, and receive reward Rt

The reward depends only on the action taken (it is i.i.d)

$$q_*(a) \doteq \mathbb{E}[R_t | A_t = a], \quad \forall a \in \{1, \ldots, k\}$$ *true values*

These true values are unknown and the distribution is unknown

Goal: Maximize your total reward

To achieve this goal you need to try actions to learn their values (**explore**) and prefer those that appear best (**exploit**)

Exploration/Exploitation Tradeoff

- Assume you had estimates

$$Q_t(a) \approx q_*(a), \quad \forall a$$ *action-value estimates*

- Define the greedy action as

$$A_t^* \doteq \arg\max_a Q_t(a)$$

If $A_t = A_t^*$ then you are exploiting

If $A_t \neq A_t^*$ then you are exploring

You can't do both but you need to do both

Action-Value Methods

- Methods that learn action-value estimates and nothing else
- Estimate action values as sample averages:

$$Q_t(a) \doteq \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbf{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbf{1}_{A_i=a}}$$

The same average estimate will converge to true values if the action is taken an infinite number of times.

$$\lim_{N_t(a)\to\infty} Q_t(a) = q_*(a)$$

The number of times action a has been taken by time t

ϵ-Greedy Action Selection

- In greedy action selection you always exploit
- In ϵ −greedy you are usually greedy but with probability ϵ you pick an action at random

- This is a simple way of balancing exploration and exploitation

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Repeat forever:

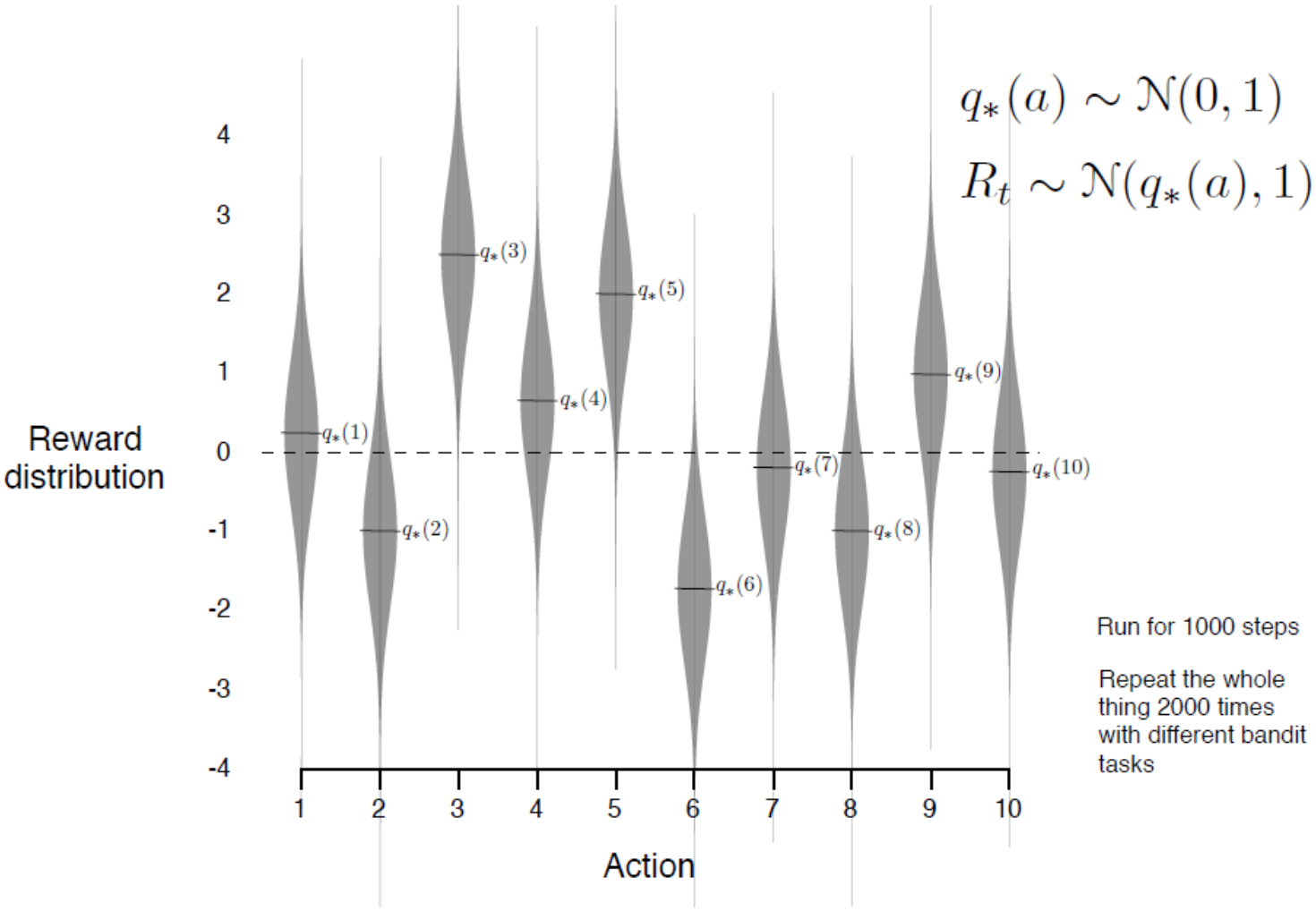
$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$R \leftarrow bandit(A)$

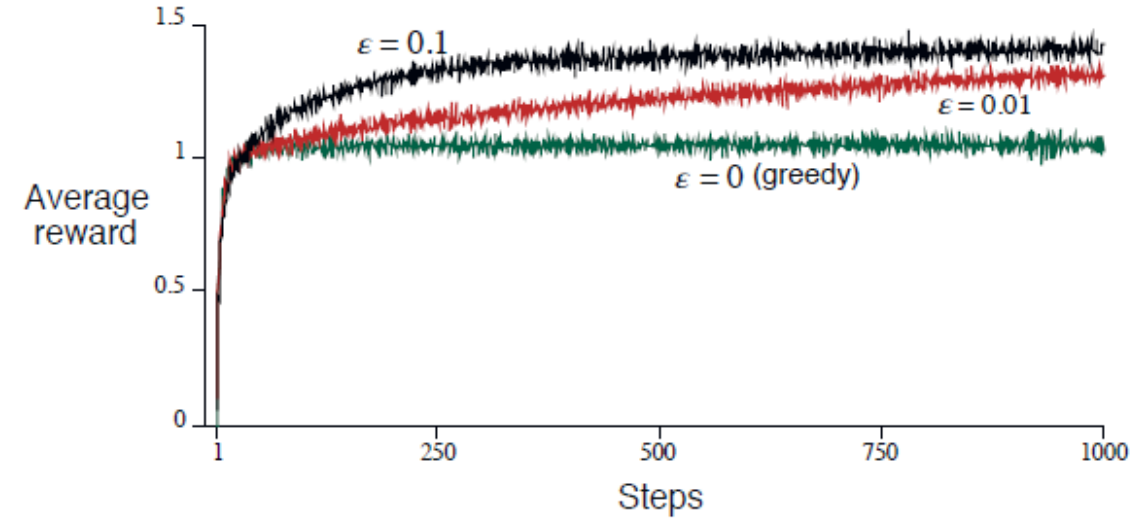
$N(A) \leftarrow N(A) + 1$

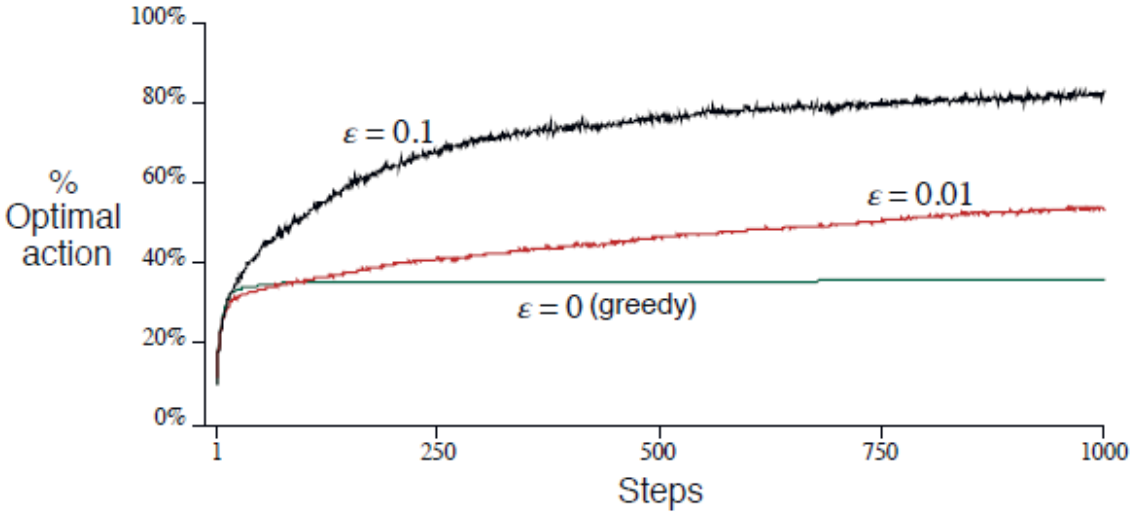
$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}\big[R - Q(A)\big]$

A Ten-Armed Example (Sutton and Barto)



ϵ-Greedy Action Selection





A simple bandit algorithm

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Repeat forever:

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

$R \leftarrow bandit(A)$

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

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

Averaging and Learning Rules

- Note that our estimate of taking some action is the average reward collected by taking that action

$$Q_n \doteq \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$$

- Can we compute this incrementally without storing all rewards to date?

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

- Standard form for learning/update rules

$$NewEstimate \leftarrow OldEstimate \; + \; StepSize \; \Big[Target - OldEstimate\Big]$$

Derivation of the Update Rule

$$Q_n \doteq \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$$

$$\begin{aligned}
Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\
&= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\Big(R_n + (n-1)Q_n\Big) \\
&= \frac{1}{n}\Big(R_n + nQ_n - Q_n\Big) \\
&= Q_n + \frac{1}{n}\Big[R_n - Q_n\Big],
\end{aligned}$$

Problems with non-stationarity

- What happens when the problem is non-stationary? (i.e. the true values change over time?)
  - Sample average is not an appropriate technique

- Instead, exponential recency-weighted average

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

where α is a constant *step-size parameter*, $\alpha \in (0, 1]$

Convergence Conditions

- To ensure convergence with probability 1, we require

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

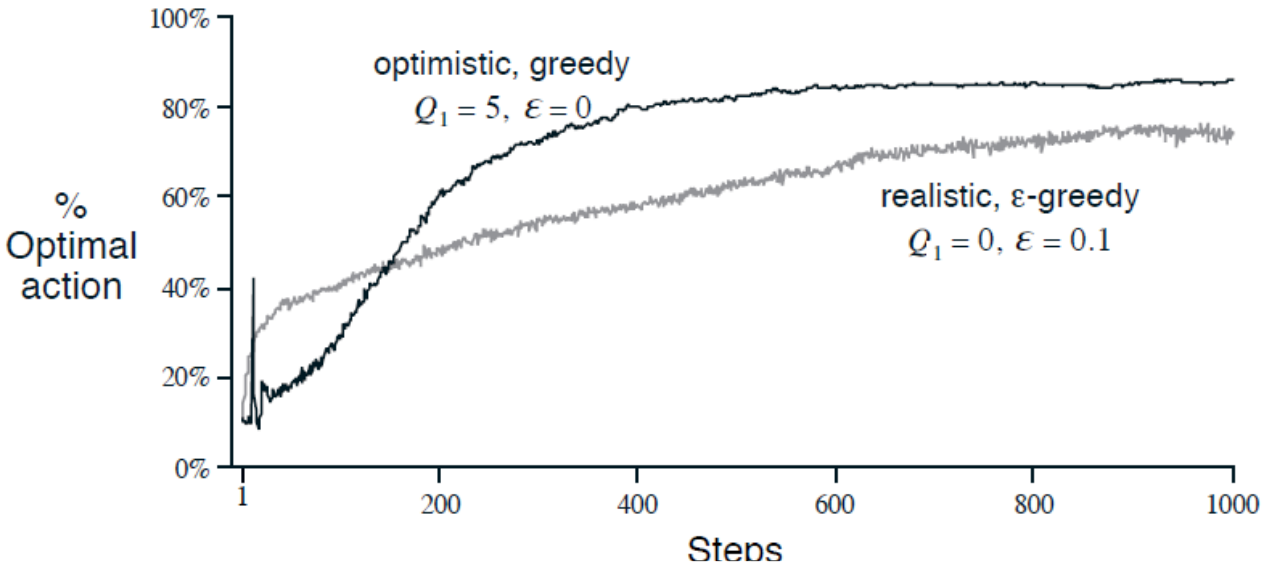
- e.g., $\alpha_n \doteq \frac{1}{n}$
- not $\alpha_n \doteq \frac{1}{n^2}$

if $\alpha_n \doteq n^{-p}, \quad p \in (0, 1)$
then convergence is
at the optimal rate:
$O(1/\sqrt{n})$

Optimism and Bandits

- We need to start somewhere with our initial values, $Q_1(a)$. So far we have assumed $Q_1(a)=0$
  - This introduces a bias

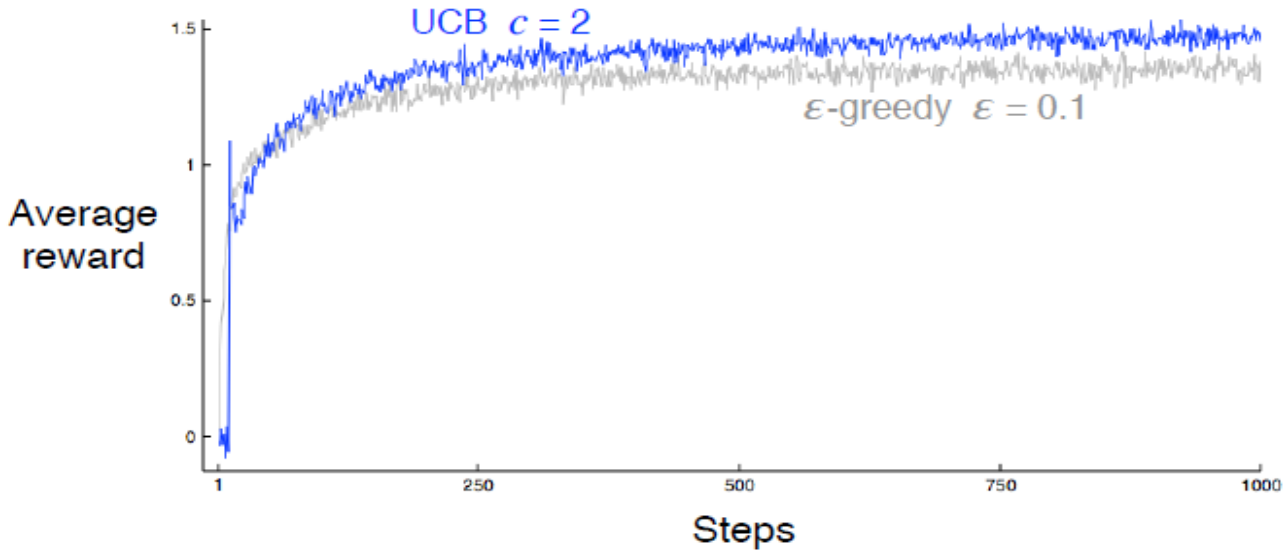
- We could be optimistic and initialize actions differently ($Q_1(a)=5$)



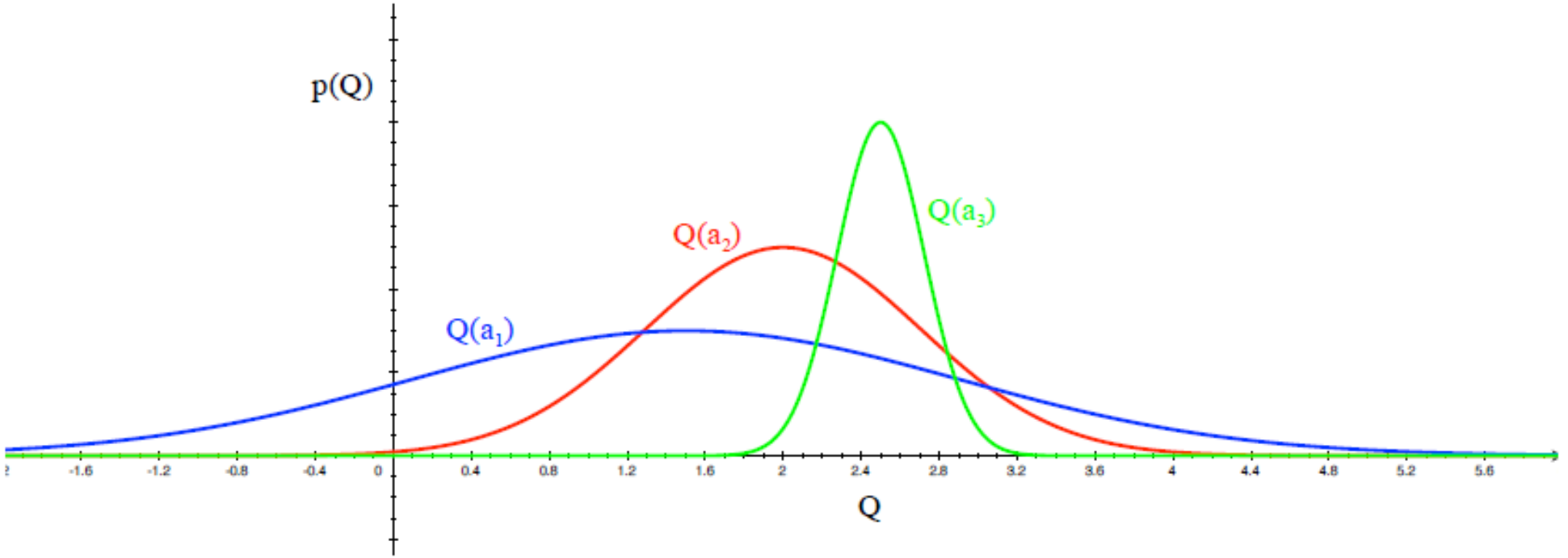
Upper Confidence Bounds

- We can also reduce exploration over time by using optimism.
- We estimate an upper bound on the true action value and then select action with largest estimated upper bound.

$$A_t \doteq \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\log t}{N_t(a)}} \right]$$



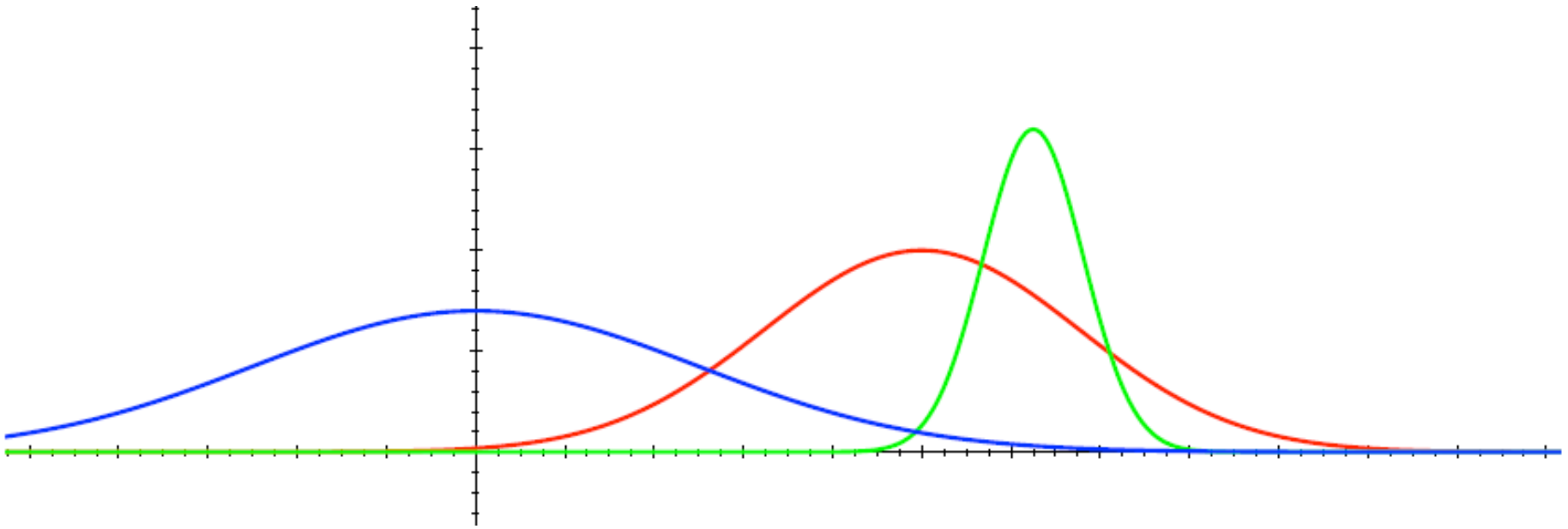
Optimism in the Face of Uncertainty



What action should be picked?

The more uncertain we are about an action-value, the more important it is to explore it since it could turn out to be the best action

Optimism in the Face of Uncertainty



After we have picked the blue action, we are more confident in its value. We are now more likely to pick another action.

Upper Confidence Bounds

- We want to estimate an upper confidence $U_t(a)$ for each action such that $q^*(a) \leq Q_t(a) + U_t(a)$

- This depends on the number of times we have sampled action a
  - Small $N_t(a)$ -> large $U_t(a)$ (estimated value is inaccurate)
  - Large $N_t(a)$ -> small $U_t(a)$ (estimated value is accurate)

- Select action maximizing Upper Confidence Bound

$$A_t = \arg\max \left[Q_t(a) + U_t(a) \right]$$

How to Determine the Bound

Theorem (Hoeffding's Inequality)

Let $X_1, \ldots, X_t$ 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

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

Apply bound to bandit awards

$$P[q^*(a) > Q_t(a) + U_t(a)] \leq e^{-2N_t(a)U_t(a)^2}$$

How to Compute the UCB

Now we pick a probability p that exceeds the value of the UCB and 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. $p=t^{-4}$

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

Summary

- Bandits are the simplest RL model
- Key challenge is the balance between exploration and exploitation