Class 3: Multi-Arm Bandit

Sutton and Barto, Chapter 2

Sutton slides and Silver

Multi-Arm Bandits

Sutton and Barto, Chapter 2

The simplest
reinforcement learning
problem

The Exploration/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

Examples

Restaurant Selection

Exploitation Go to your favourite restaurant

Exploration Try a new restaurant

Online Banner Advertisements

Exploitation Show the most successful advert

Exploration Show a different advert

Oil Drilling

Exploitation Drill at the best known location

Exploration Drill at a new location

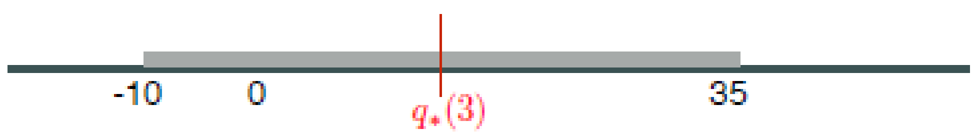
Game Playing

Exploitation Play the move you believe is best

Exploration Play an experimental move

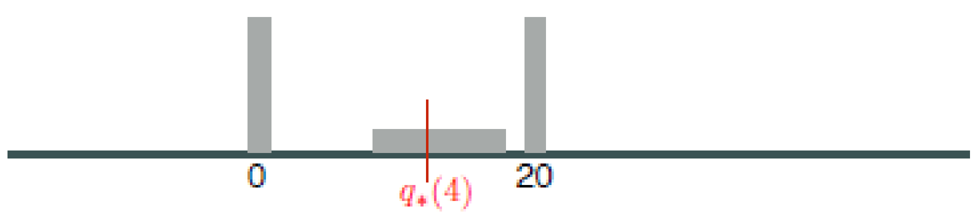
You are the algorithm! (bandit1)

- Action 1 — Reward is always 8
  - value of action 1 is $q_*(1) =$
- Action 2 — 88% chance of 0, 12% chance of 100!
  - value of action 2 is $q_*(2) = .88 \times 0 + .12 \times 100 =$
- Action 3 — Randomly between -10 and 35, equiprobable



$q_*(3) =$

- Action 4 — a third 0, a third 20, and a third from {8,9,…, 18}



The *k*-armed Bandit Problem

- On each of a sequence of *time steps*, t=1,2,3,…,
  you choose an action A_t from k possibilities, and receive a real-valued *reward* R_t

- The reward depends only on the action taken;
  it is indentically, independently distributed (i.i.d.):

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

true values

- These true values are *unknown*. The distribution is unknown

- Nevertheless, you must maximize your total reward

- You must both try actions to learn their values (explore), and prefer those that appear best (exploit)

The Exploration/Exploitation Dilemma

- Suppose you form estimates

$$Q_t(a) \approx q_*(a), \quad \forall a$$

action-value estimates

- Define the *greedy action* at time *t* 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

- You can never stop exploring, but maybe you should explore less with time. Or maybe not.

Regret

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

- $q^*(a) = E[r \mid a]$

The *optimal value* V^* is

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

The *regret* is the opportunity loss for one step

- $l_t = E[V^* - Q(a_t)]$

The *total regret* is the total opportunity loss

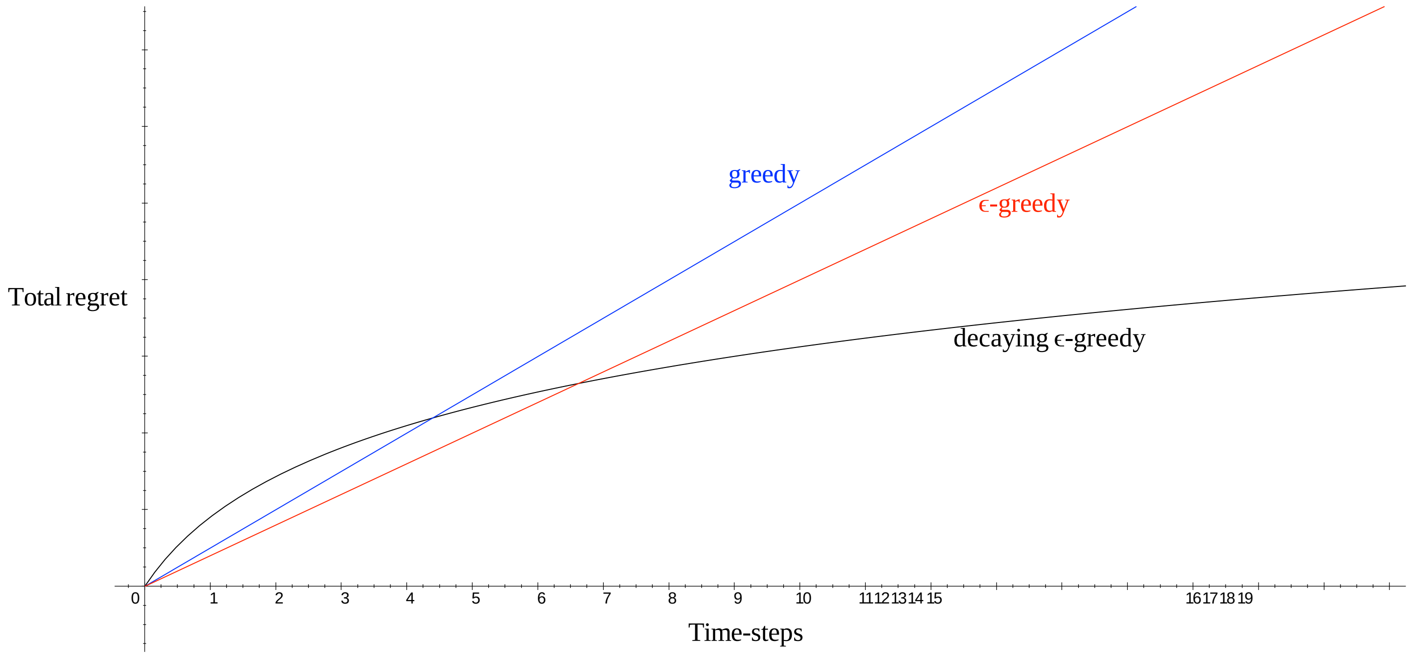
$$L_t = \mathbb{E}\left[\sum_{\tau=1}^{t} V^* - Q(a_\tau)\right]$$

- Maximise cumulative reward $\equiv$ minimise total regret

- The *count* $N_t(a)$ is expected number of selections for action a
- 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!



- 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 sublinear total regret?

Complexity of regret

- The performance of any algorithm is determined by similarity between optimal arm and other arms
- Hard problems have similar-looking arms with different means
- This is described formally by the gap Δ_a and the similarity in distributions $KL(\mathcal{R}^a || \mathcal{R}^a *)$

Theorem (Lai and Robbins)

Asymptotic total regret is at least logarithmic in number of steps

$$\lim_{t \to \infty} L_t \geq \log t \sum_{a | \Delta_a > 0} \frac{\Delta_a}{KL(\mathcal{R}^a || \mathcal{R}^{a^*})}$$

Overview

- Action-value methods
  - Epsilon-greedy strategy
  - Incremental implementation
  - Stationary vs. non-stationary environment
  - Optimistic initial values
- UCB action selection
- Gradient bandit algorithms
- Associative search (contextual bandits)

Basics

- Maximize total reward collected
  - vs learn (optimal) policy (RL)
- Episode is one step
- Complex function of
  - True value
  - Uncertainty
  - Number of time steps
  - Stationary vs non-stationary?

Action-Value Methods

- Methods that learn action-value estimates and nothing else
- For example, 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 sample-average estimates converge to the 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 instead pick an action at random (possibly the greedy action again)
- This is perhaps the simplest way to balance exploration and exploitation

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Repeat forever:

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

$R \leftarrow bandit(A)$

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

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

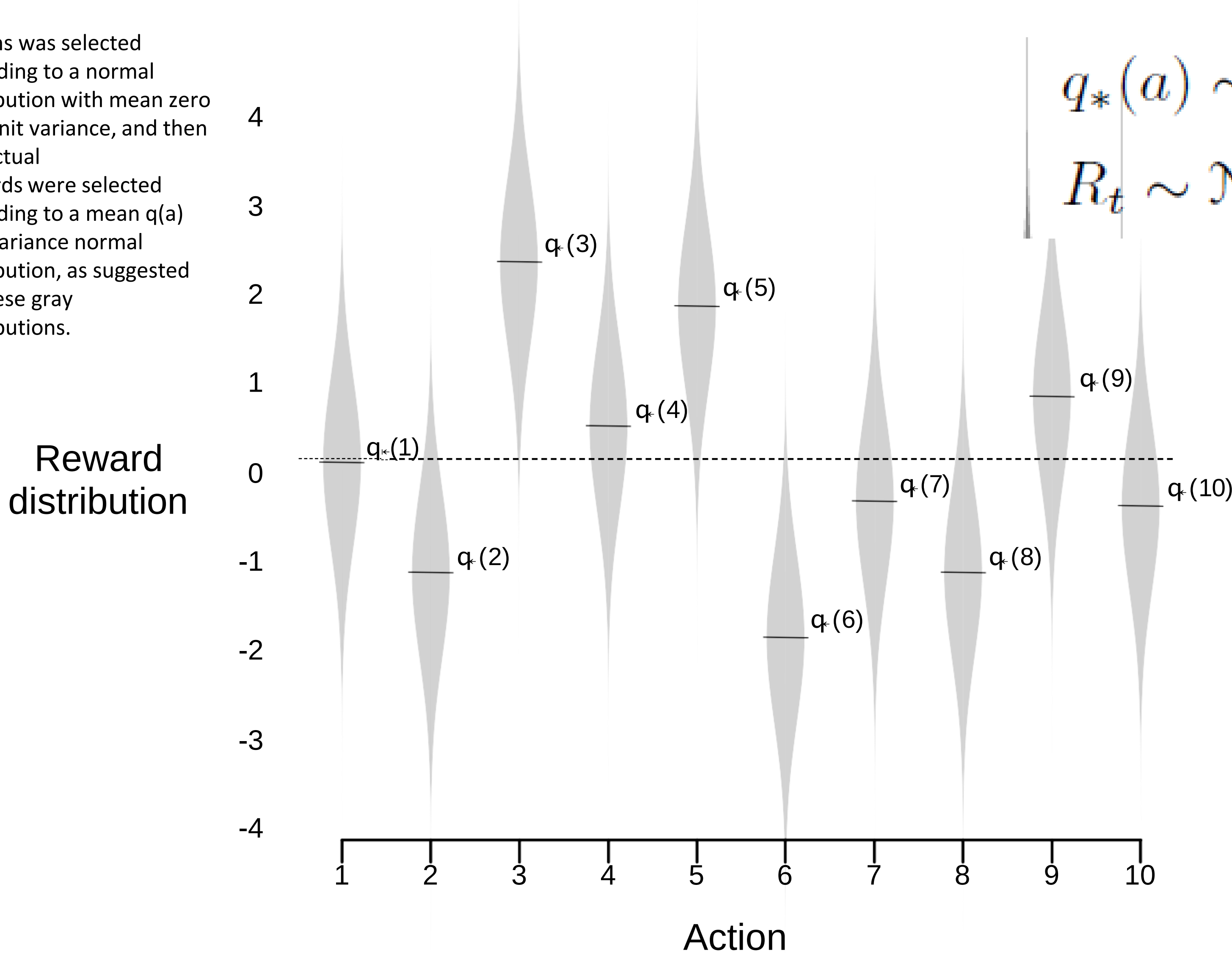
One Bandit Task from

The 10-armed Testbed

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.

$$q_*(a) \sim \mathcal{N}(0, 1)$$

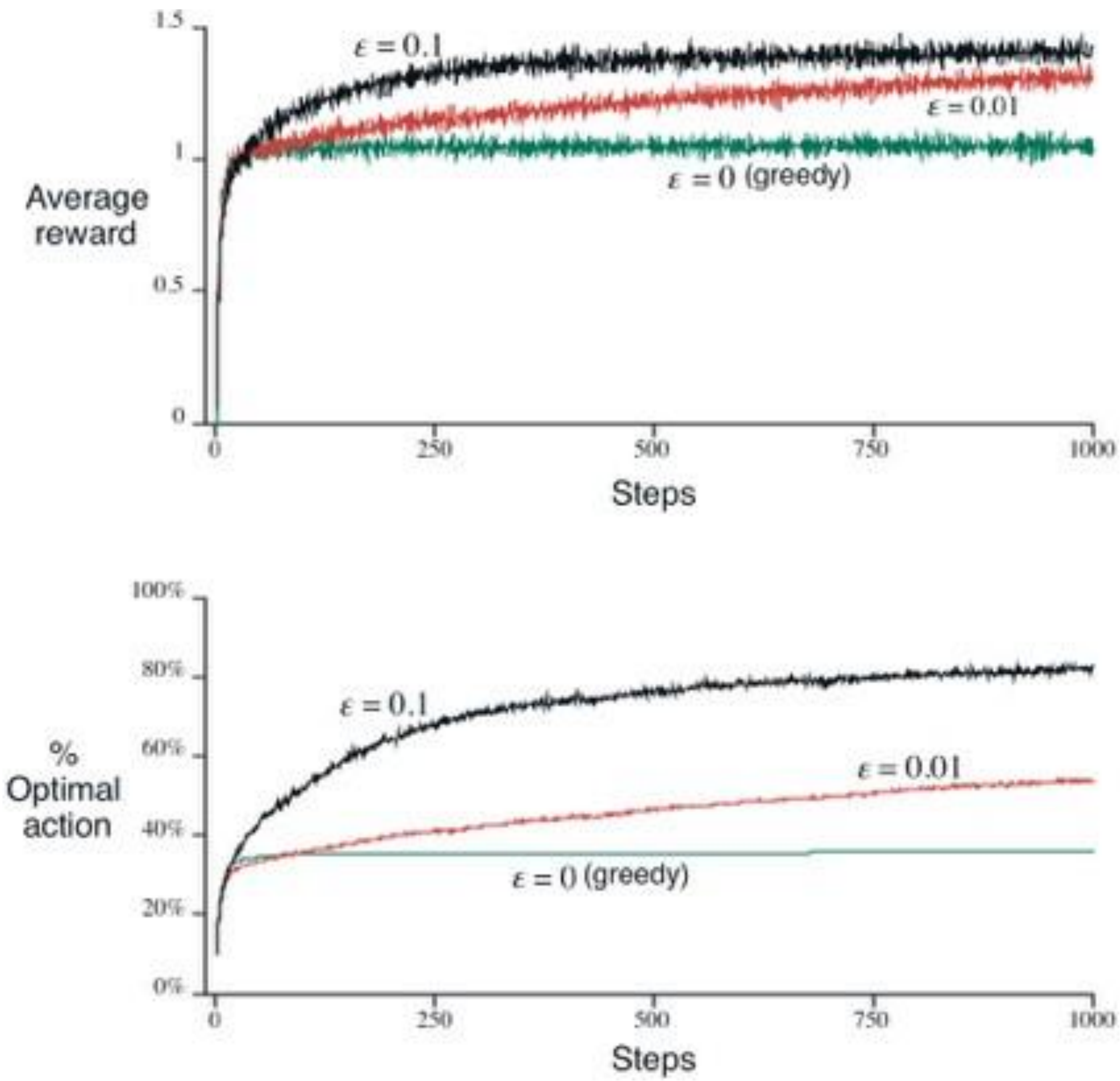
$$R_t \sim \mathcal{N}(q_*(a), 1)$$



Run for 1000 steps

Repeat the whole thing 2000 times with different bandit tasks

ε-Greedy Methods on the 10-Armed Testbed



Averaging ⟶ learning rule

- To simplify notation, let us focus on one action
  - We consider only its rewards, and its estimate after $n+1$ rewards:

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

- How can we do this incrementally (without storing all the rewards)?
- Could store a running sum and count (and divide), or equivalently:

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

- This is a standard form for learning/update rules:

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

Derivation of incremental update

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

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

Tracking a Non-stationary Problem

- Suppose the true action values change slowly over time
  - then we say that the problem is *nonstationary*
- In this case, sample averages are not a good idea (Why?)
- Better is an "exponential, recency-weighted average":

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

where α is a constant, *step-size parameter*, $0 < \alpha \leq 1$

- There is bias due to Q_1 that becomes smaller over time

Standard stochastic approximation convergence conditions

- To assure convergence with probability 1:

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

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

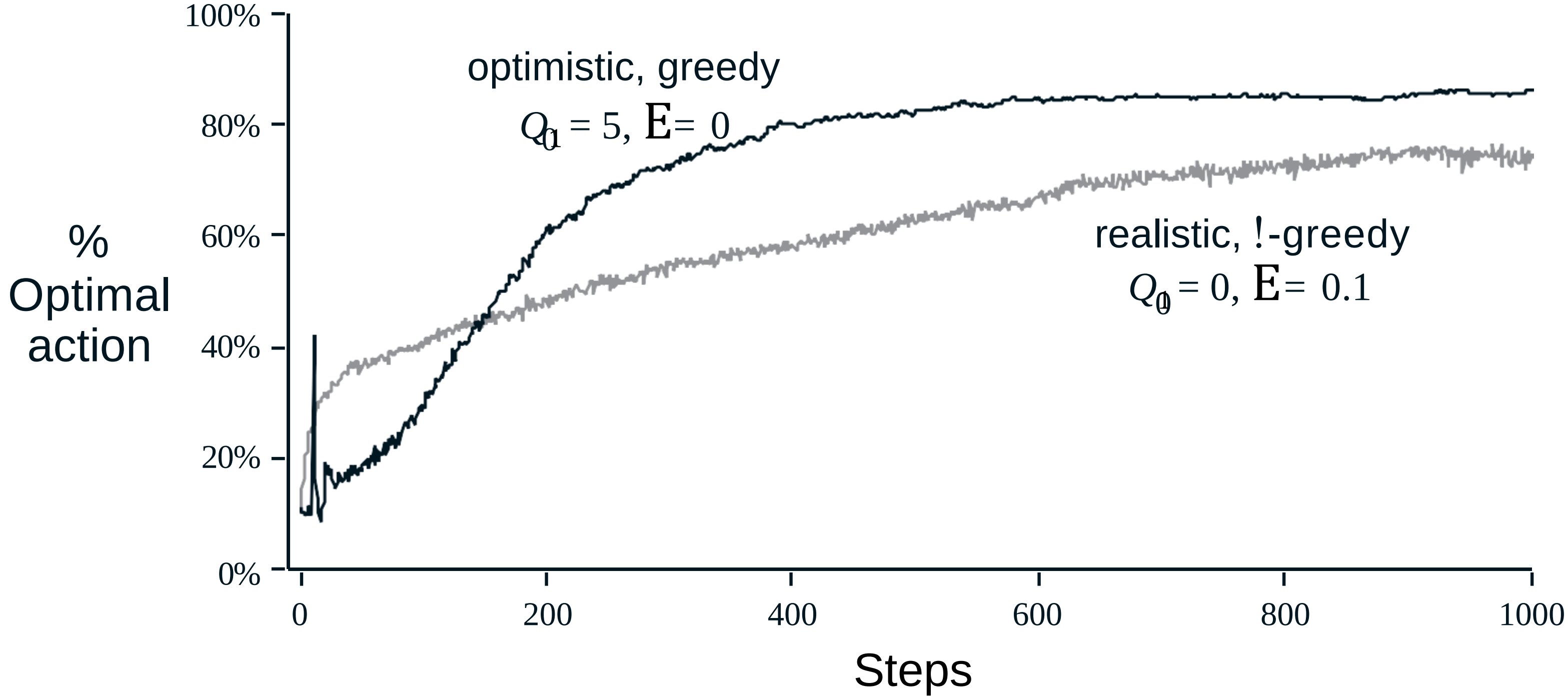
if $\alpha_n = n^{-p}, \quad p \in (0,1)$
then convergence is at the optimal rate:

$$O(1/\sqrt{n})$$

Optimistic Initial Values

- All methods so far depend on $Q_1(a)$, i.e.,they are biased. So far we have used $Q_1(a) = 0$

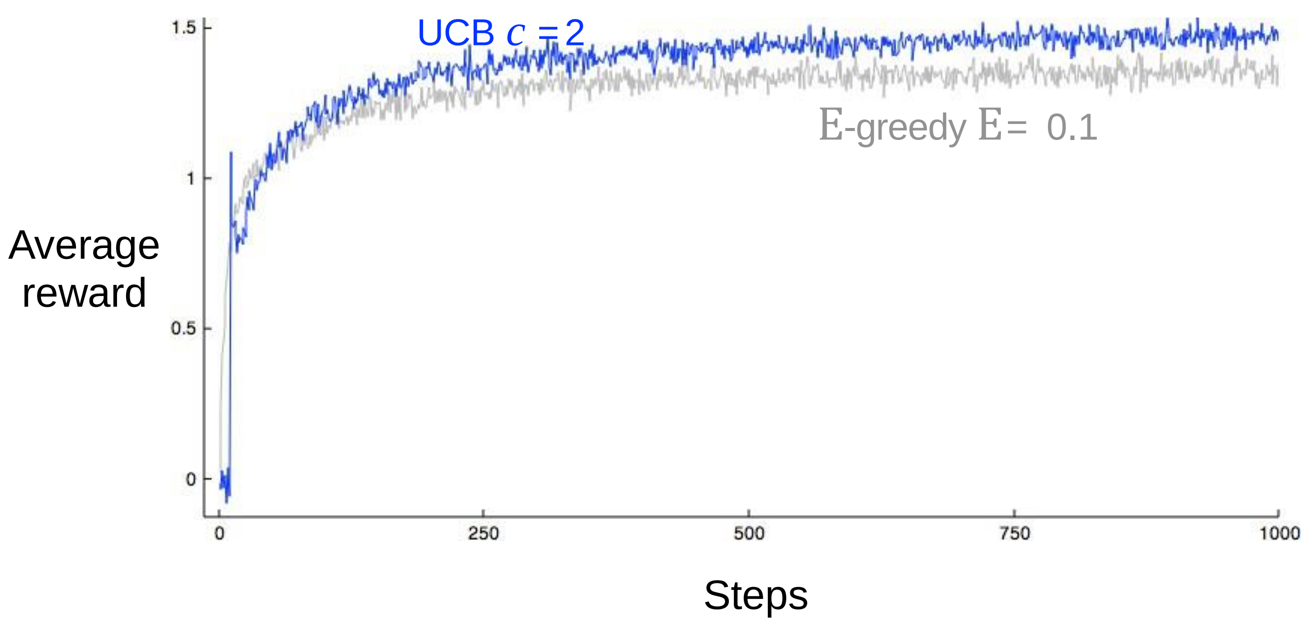
- Suppose we initialize the action values *optimistically* ($Q_1(a) = 5$), e.g., on the 10-armed testbed (with alpha= 0.1)



Upper Confidence Bound (UCB) action selection

- A clever way of reducing exploration over time
- Focus on actions whose estimate has large degree of uncertainty
- Estimate an upper bound on the true action values
- Select the action with the largest (estimated) upper bound

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



Complexity of UCB Algorithm

Theorem

The UCB algorithm achieves logarithmic asymptotic total regret

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

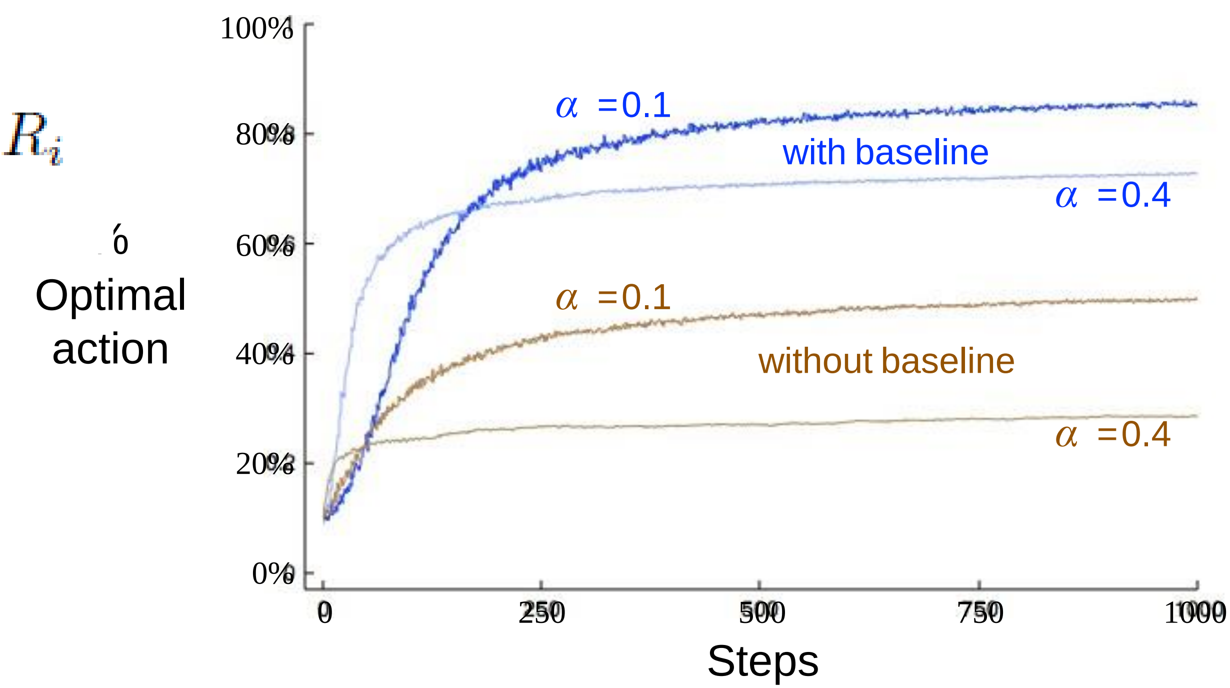
Gradient-Bandit Algorithms

- Let $H_t(a)$ be a learned *preference* for taking action a

$$\Pr\{A_t = a\} \doteq \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}} \doteq \pi_t(a)$$

$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha\big(R_t - \bar{R}_t\big)\big(1 - \pi_t(A_t)\big), \quad \text{and}$$
$$H_{t+1}(a) \doteq H_t(a) - \alpha\big(R_t - \bar{R}_t\big)\pi_t(a), \quad \text{for all } a \neq A_t, \tag{2.10}$$

$$\bar{R}_t \doteq \frac{1}{t}\sum_{i=1}^{t} R_i$$



Derivation of gradient-bandit algorithm

In exact *gradient ascent*:

$$H_{t+1}(a) \doteq H_t(a) + \alpha \frac{\partial \mathbb{E}\,[R_t]}{\partial H_t(a)}, \qquad (1)$$

where:

$$\mathbb{E}[R_t] \doteq \sum_b \pi_t(b) q_*(b),$$

$$\begin{aligned}
\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} &= \frac{\partial}{\partial H_t(a)} \left[\sum_b \pi_t(b) q_*(b) \right] \\
&= \sum_b q_*(b) \frac{\partial \pi_t(b)}{\partial H_t(a)} \\
&= \sum_b \left(q_*(b) - X_t \right) \frac{\partial \pi_t(b)}{\partial H_t(a)},
\end{aligned}$$

where X_t does not depend on b, because $\sum_b \frac{\partial \pi_t(b)}{\partial H_t(a)} = 0$.

$$\frac{\partial\, \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_b \left(q_*(b) - X_t\right) \frac{\partial\, \pi_t(b)}{\partial H_t(a)}$$

$$= \sum_b \pi_t(b)\left(q_*(b) - X_t\right) \frac{\partial\, \pi_t(b)}{\partial H_t(a)} / \pi_t(b)$$

$$= \mathbb{E}\left[\left(q_*(A_t) - X_t\right) \frac{\partial\, \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right]$$

$$= \mathbb{E}\left[\left(R_t - \bar{R}_t\right) \frac{\partial\, \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right],$$

where here we have chosen $X_t = \bar{R}_t$ and substituted R_t for $q_*(A_t)$, which is permitted because $\mathbb{E}[R_t | A_t] = q_*(A_t)$.

For now assume: $\frac{\partial\, \pi_t(b)}{\partial H_t(a)} = \pi_t(b)\left(\mathbf{1}_{a=b} - \pi_t(a)\right)$. Then:

$$= \mathbb{E}\left[\left(R_t - \bar{R}_t\right)\pi_t(A_t)\left(\mathbf{1}_{a=A_t} - \pi_t(a)\right)/\pi_t(A_t)\right]$$

$$= \mathbb{E}\left[\left(R_t - \bar{R}_t\right)\left(\mathbf{1}_{a=A_t} - \pi_t(a)\right)\right].$$

$$H_{t+1}(a) = H_t(a) + \alpha\left(R_t - \bar{R}_t\right)\left(\mathbf{1}_{a=A_t} - \pi_t(a)\right), \text{ (from (1), QED)}$$

Thus it remains only to show that

$$\frac{\partial\, \pi_t(b)}{\partial H_t(a)} = \pi_t(b)\big(\mathbf{1}_{a=b} - \pi_t(a)\big).$$

Recall the standard quotient rule for derivatives:

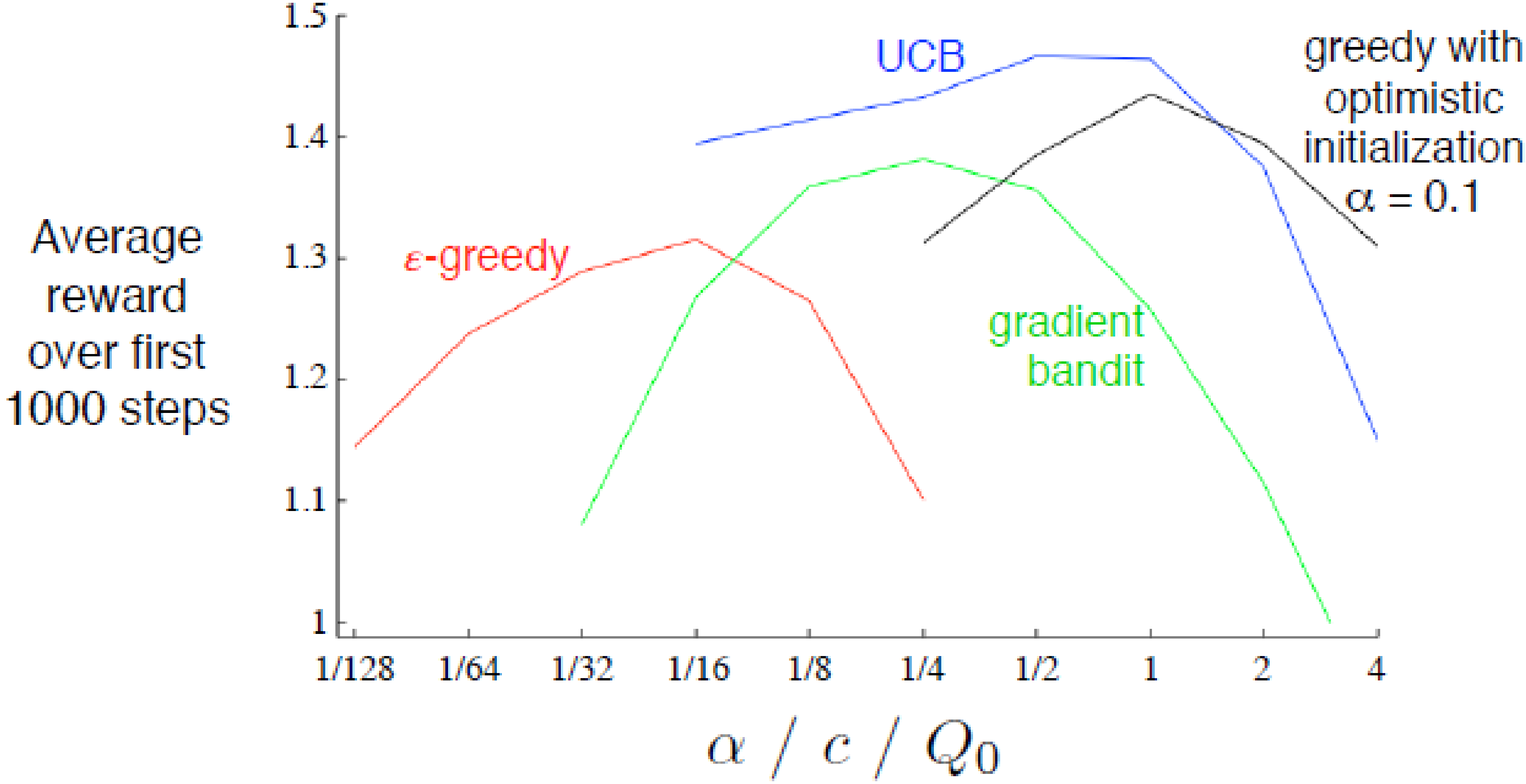
$$\frac{\partial}{\partial x}\left[\frac{f(x)}{g(x)}\right] = \frac{\frac{\partial f(x)}{\partial x} g(x) - f(x)\frac{\partial g(x)}{\partial x}}{g(x)^2}.$$

Using this, we can write...

Quotient Rule: $\frac{\partial}{\partial x}\left[\frac{f(x)}{g(x)}\right] = \frac{\frac{\partial f(x)}{\partial x} g(x) - f(x)\frac{\partial g(x)}{\partial x}}{g(x)^2}$

$$
\begin{aligned}
\frac{\partial \pi_t(b)}{\partial H_t(a)} &= \frac{\partial}{\partial H_t(a)} \pi_t(b) \\
&= \frac{\partial}{\partial H_t(a)} \left[\frac{e^{h_t(b)}}{\sum_{c=1}^{k} e^{h_t(c)}}\right] \\
&= \frac{\frac{\partial e^{h_t(b)}}{\partial H_t(a)} \sum_{c=1}^{k} e^{h_t(c)} - e^{h_t(b)} \frac{\partial \sum_{c=1}^{k} e^{h_t(c)}}{\partial H_t(a)}}{\left(\sum_{c=1}^{k} e^{h_t(c)}\right)^2} && \text{(Q.R.)} \\
&= \frac{\mathbf{1}_{a=b} e^{h_t(a)} \sum_{c=1}^{k} e^{h_t(c)} - e^{h_t(b)} e^{h_t(a)}}{\left(\sum_{c=1}^{k} e^{h_t(c)}\right)^2} && \left(\frac{\partial e^x}{\partial x} = e^x\right) \\
&= \frac{\mathbf{1}_{a=b} e^{h_t(b)}}{\sum_{c=1}^{k} e^{h_t(c)}} - \frac{e^{h_t(b)} e^{h_t(a)}}{\left(\sum_{c=1}^{k} e^{h_t(c)}\right)^2} \\
&= \mathbf{1}_{a=b} \pi_t(b) - \pi_t(b)\pi_t(a) \\
&= \pi_t(b)\left(\mathbf{1}_{a=b} - \pi_t(a)\right). && \text{(Q.E.D.)}
\end{aligned}
$$

Summary Comparison of Bandit Algorithms



Conclusions

- These are all simple methods
  - but they are complicated enough—we will build on them
  - we should understand them completely
  - there are still open questions
- Our first algorithms that learn from evaluative feedback
  - and thus must balance exploration and exploitation
- Our first algorithms that appear to have a goal
  —that learn to maximize reward by trial and error