

Deep Reinforcement Learning and Control

# Exploration/Exploitation in Multi-armed Bandits

Spring 2019, CMU 10-403

Katerina Fragkiadaki



# Used Materials

- **Disclaimer:** Some of the material and slides for this lecture were borrowed from Russ Salakhutdinov who in turn borrowed from Rich Sutton's class and David Silver's class on Reinforcement Learning.

# Supervised VS Reinforcement Learning

- Supervised learning (instructive feedback): the expert directly suggests correct actions
- Learning by interaction (evaluative feedback): the environment provides signal whether actions the agent selects are good or bad, not even how far away they are from the optimal actions!
- Evaluative feedback depends on the current policy the agent has
- Exploration: active search for good actions to execute

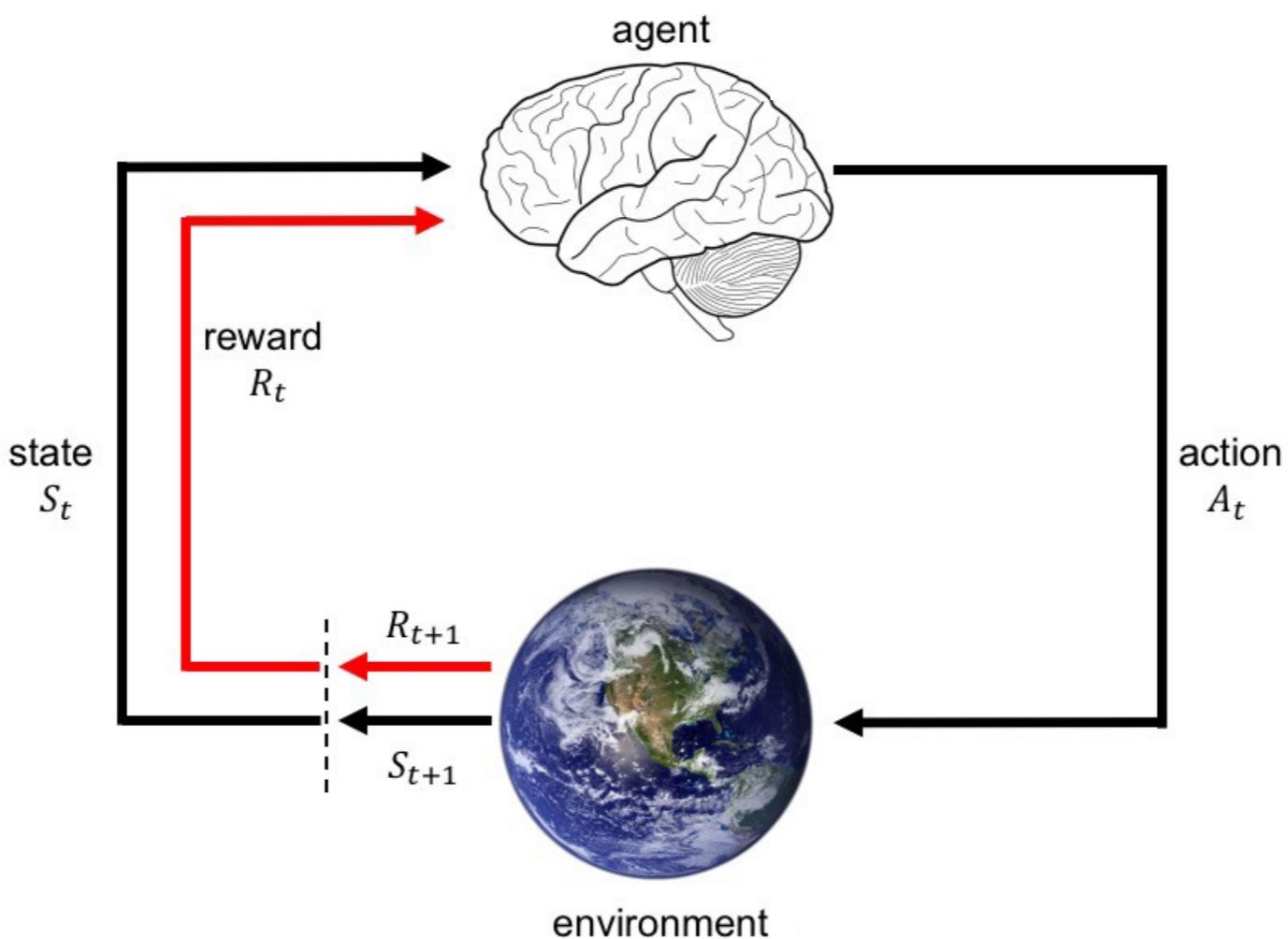
# Exploration vs. Exploitation Dilemma

- ▶ 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

# Reinforcement learning



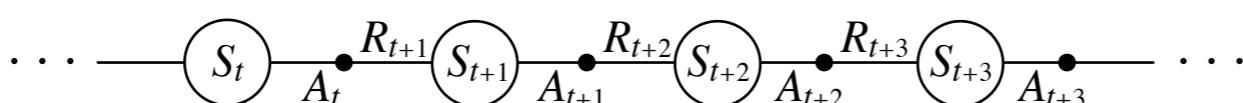
Agent and environment interact at discrete time steps:  $t = 0, 1, 2, 3, \dots$

Agent observes state at step  $t$ :  $S_t \in \mathcal{S}$

produces action at step  $t$ :  $A_t \in \mathcal{A}(S_t)$

gets resulting reward:  $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$

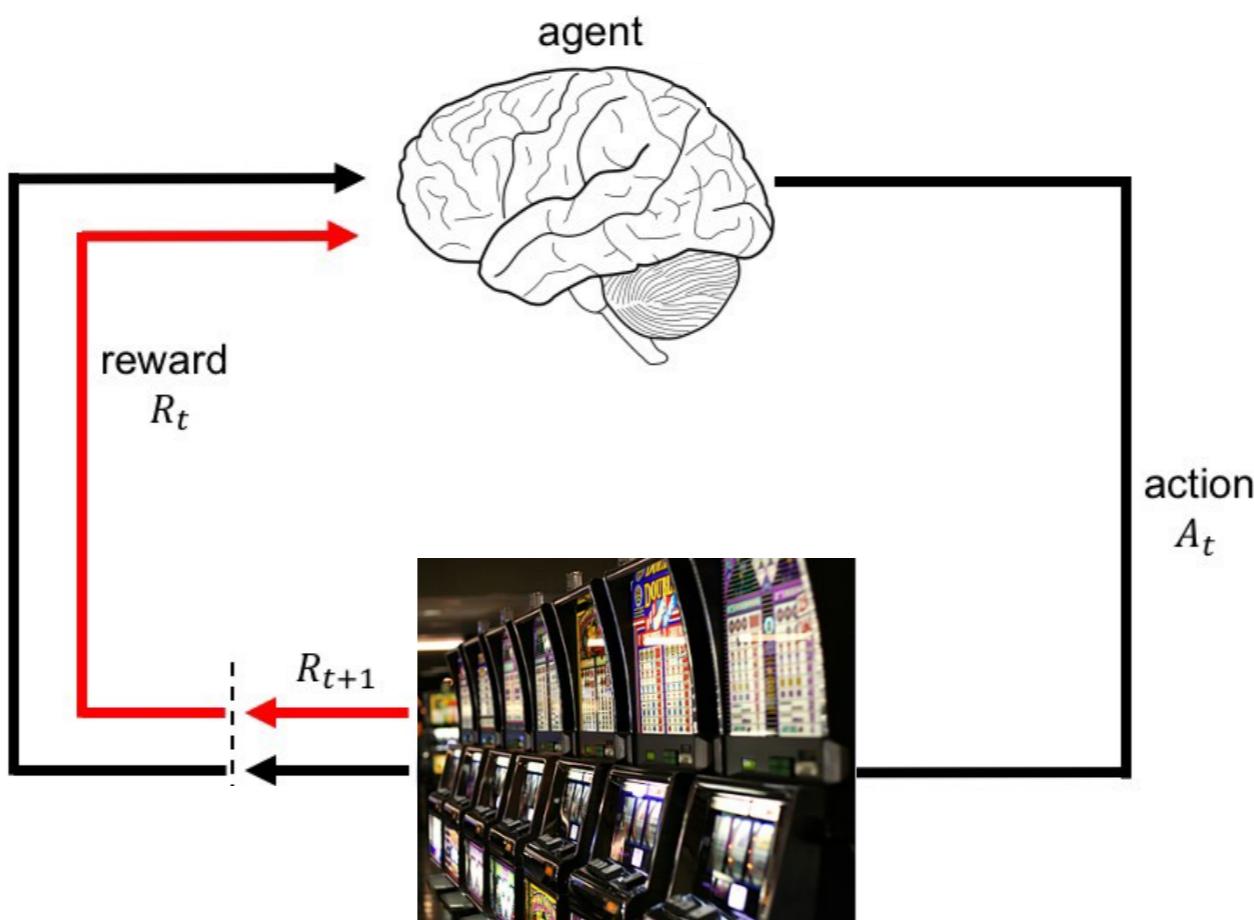
and resulting next state:  $S_{t+1} \in \mathcal{S}^+$



# This lecture

A closer look to exploration-exploitation balancing in a simplified RL setup

# Multi-Armed Bandits



$$A_t, R_{t+1}, A_{t+1}, R_{t+2}, A_{t+2}, A_{t+3}, R_{t+3}, \dots$$

The state does not change.

# Multi-Armed Bandits

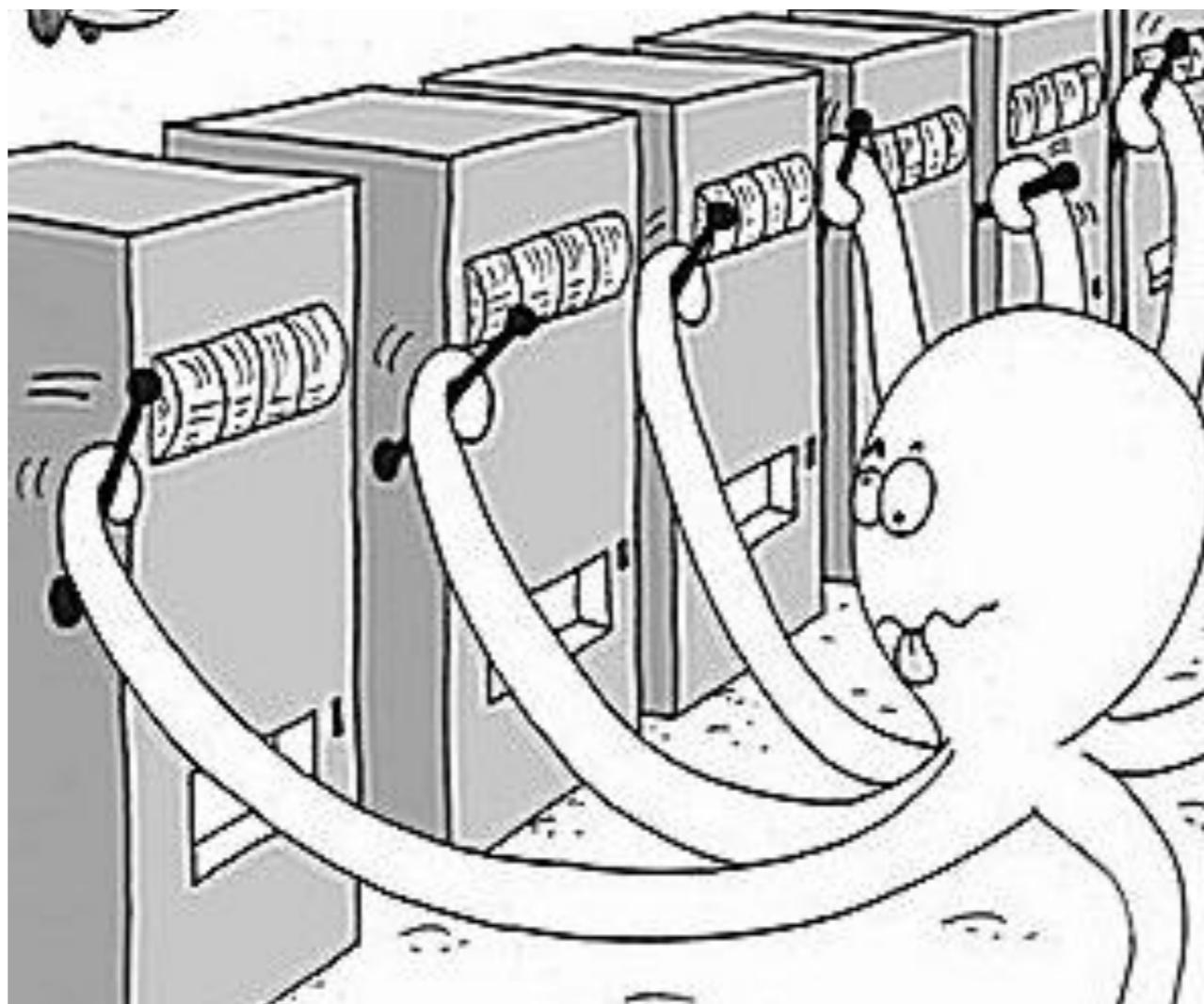
One-armed bandit= Slot machine (English slang)



source: [infoslotmachine.com](http://infoslotmachine.com)

# Multi-Armed Bandits

- Multi-Armed bandit = Multiple Slot Machine



source: Microsoft Research

# Multi-Armed Bandit Problem

At each timestep  $t$  the agent chooses one of the  $K$  arms and plays it.

The  $i$ th arm produces reward  $r_{i,t}$  when played at timestep  $t$ .

The rewards  $r_{i,t}$  are drawn from a probability distribution  $\mathcal{P}_i$  with mean  $\mu_i$

The agent **does not know neither the arm reward distributions neither their means**



source: Pandey et al.'s slide

Alternative notation for mean arm rewards:

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

Agent's Objective:

- Maximize cumulative rewards.
- In other words: **Find the arm with the highest mean reward**

# Example: Bernoulli Bandits

Recall: The **Bernoulli distribution** is the discrete probability distribution of a random variable which takes the value 1 with probability  $p$  and the value 0 with probability  $q=1-p$ , that is, the probability distribution of any single experiment that asks a yes–no question

- Each action (arm when played) results in success or failure. Rewards are binary!
- Mean reward for each arm represents the probability of success
- Action (arm)  $k \in \{1, \dots, K\}$  produces a success with probability  $\theta_k \in [0, 1]$ .



$$\mu_1$$



$$\mu_2$$



$$\mu_3$$

source: Pandey et al.'s slide

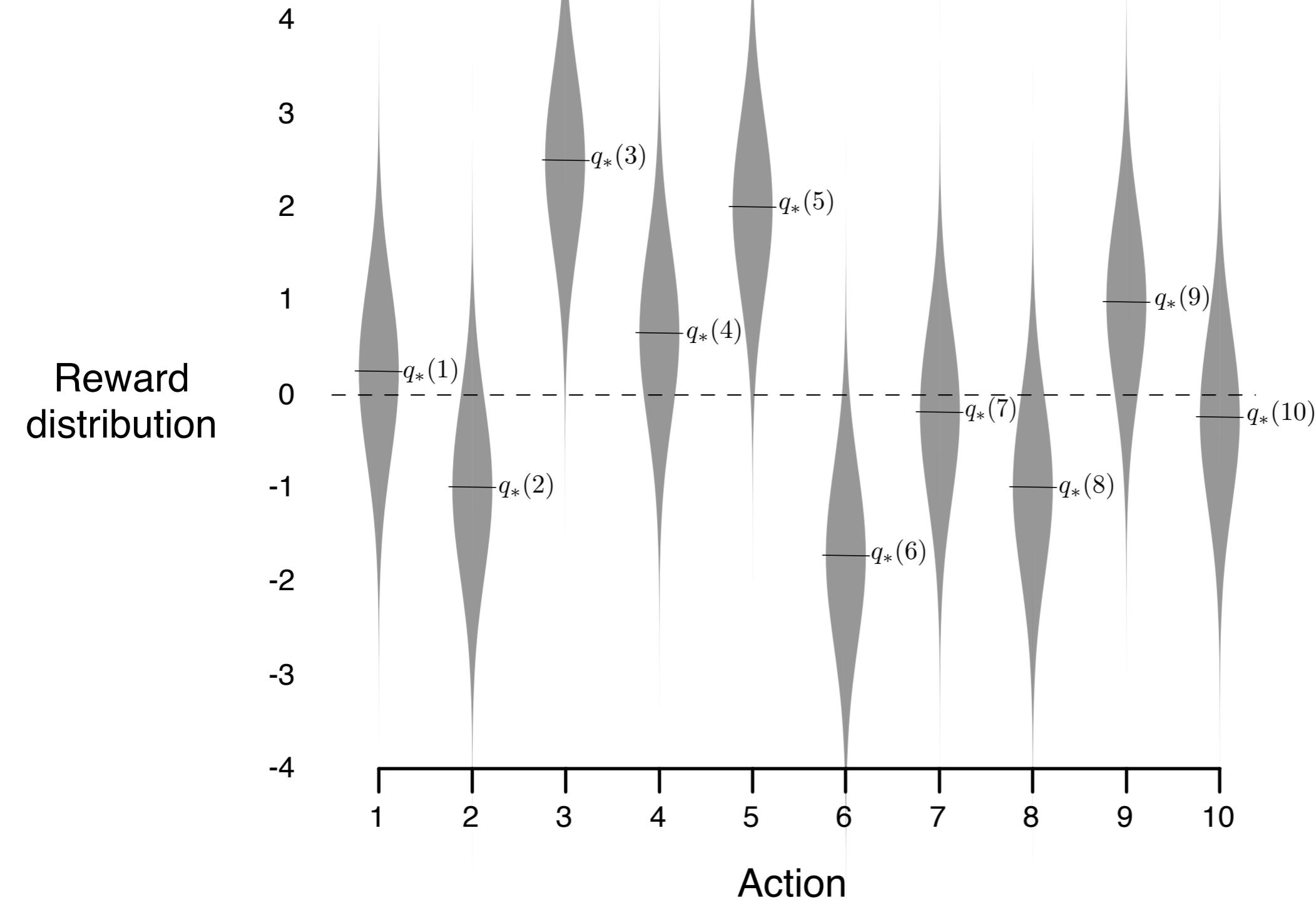
win 0.6  
of time

win 0.4  
of time

win 0.45  
of time

# Example: Gaussian Bandits

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



# Real world motivation: A/B testing

- Two arm bandits: each arm corresponds to an image variation shown to users (not necessarily the same user)
- Mean rewards: the total percentage of users that would click on each invitation

OBAMA • BIDEN

DINNER WITH BARACK  
Your chance to meet the President

GET STARTED

YOU'RE INVITED.  
WE'LL COVER YOUR AIRFARE.

No purchase, payment, or contribution necessary to enter or win. Contributing will not improve chances of winning. Void where prohibited. Entries must be received by September 20, 2012. You may enter by contributing to Obama Victory Fund 2012 here or click here to enter without contributing. Three winners will each receive the following prize package: round-trip tickets for winner from within the fifty U.S. States, DC, or Puerto Rico to a destination to be determined by the Sponsor; hotel accommodations; and dinner with President Obama on a date to be determined by the Sponsor (approximate retail value of all prizes \$4,800). Odds of winning depend on number of entries received. Promotion open only to U.S. citizens, or lawful permanent U.S. residents who are legal residents of 50 United States, District of Columbia and Puerto Rico and 18 or older (or age of majority under applicable law). Promotion subject to Official Rules. Official rules and additional restrictions on eligibility. Sponsor: Obama for America, 130 E. Randolph St., Chicago, IL 60601.

OBAMA • BIDEN

Privacy Policy Terms of Service

Contributions or gifts to Obama Victory Fund 2012 are not tax deductible.

PAID FOR BY OBAMA VICTORY FUND 2012. A JOINT FUNDRAISING COMMITTEE AUTHORIZED BY OBAMA FOR AMERICA, THE DEMOCRATIC NATIONAL COMMITTEE, AND THE STATE DEMOCRATIC PARTIES IN THE FOLLOWING STATES: CO, FL, IA, NV, NH, NC, OH, PA, VA, AND WI.

© 2011–2012 Obama for America. All Rights Reserved.

OBAMA • BIDEN

DINNER WITH BARACK  
Your chance to meet the President

GET STARTED

You're invited.  
We'll cover your airfare.

No purchase, payment, or contribution necessary to enter or win. Contributing will not improve chances of winning. Void where prohibited. Entries must be received by September 20, 2012. You may enter by contributing to Obama Victory Fund 2012 here or click here to enter without contributing. Three winners will each receive the following prize package: round-trip tickets for winner from within the fifty U.S. States, DC, or Puerto Rico to a destination to be determined by the Sponsor; hotel accommodations; and dinner with President Obama on a date to be determined by the Sponsor (approximate retail value of all prizes \$4,800). Odds of winning depend on number of entries received. Promotion open only to U.S. citizens, or lawful permanent U.S. residents who are legal residents of 50 United States, District of Columbia and Puerto Rico and 18 or older (or age of majority under applicable law). Promotion subject to Official Rules. Official rules and additional restrictions on eligibility. Sponsor: Obama for America, 130 E. Randolph St., Chicago, IL 60601.

OBAMA • BIDEN

Privacy Policy Terms of Service

Contributions or gifts to Obama Victory Fund 2012 are not tax deductible.

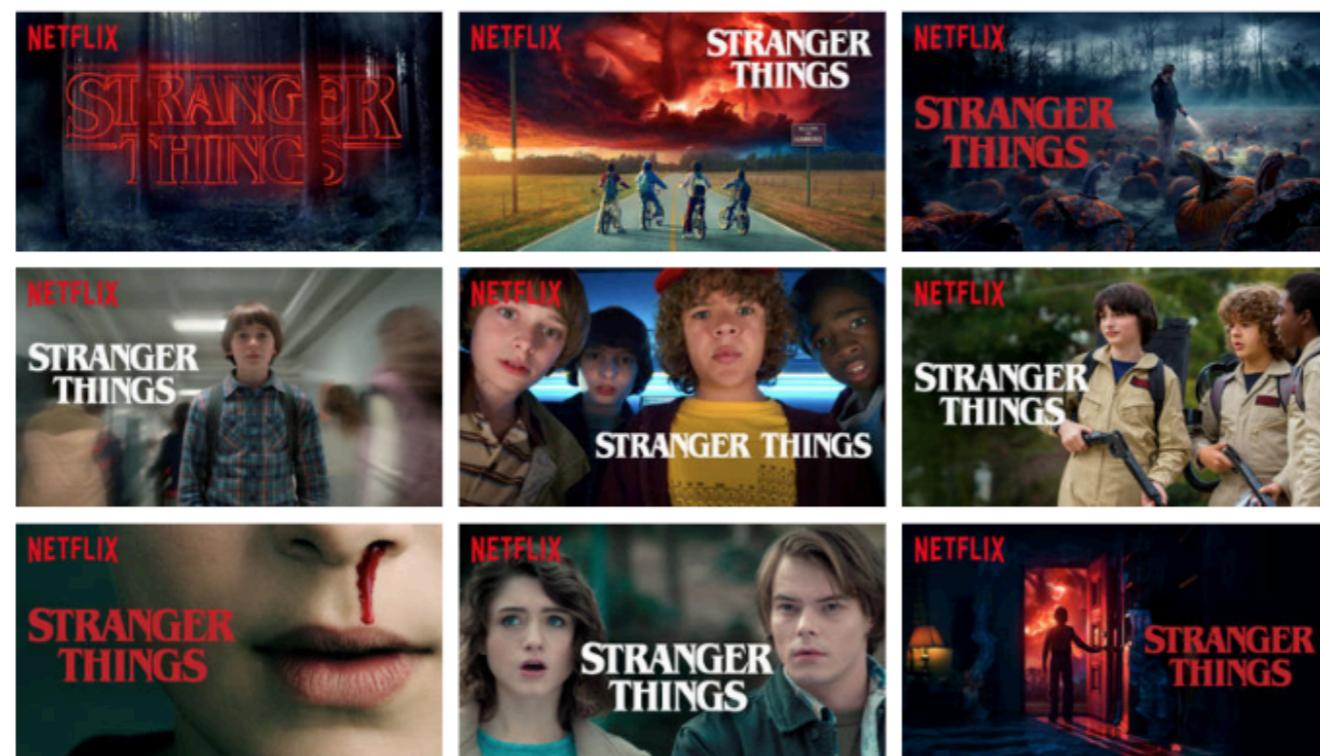
PAID FOR BY OBAMA VICTORY FUND 2012. A JOINT FUNDRAISING COMMITTEE AUTHORIZED BY OBAMA FOR AMERICA, THE DEMOCRATIC NATIONAL COMMITTEE, AND THE STATE DEMOCRATIC PARTIES IN THE FOLLOWING STATES: CO, FL, IA, NV, NH, NC, OH, PA, VA, AND WI.

© 2011–2012 Obama for America. All Rights Reserved.

# Real world motivation: NETFLIX artwork

For a particular movie, we want to decide what image to show (to all the NETFLIX users)

- Actions: uploading one of the K images to a user's home screen
- Ground-truth mean rewards (unknown): the % of NETFLIX users that will click on the title and watch the movie
- Estimated mean rewards: the average click rate observed (quality engagement, not clickbait)



# The Exploration/Exploitation Dilemma

- Suppose you form estimates

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

# The Exploration/Exploitation Dilemma

- Suppose you form estimates

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

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

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

# The Exploration/Exploitation Dilemma

- Suppose you form estimates

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

- 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*

# The Exploration/Exploitation Dilemma

- Suppose you form estimates

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

- 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

# The Exploration/Exploitation Dilemma

- Suppose you form estimates

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

- 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.

# Regret

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

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

- ▶ The optimal value is

$$\nu_* = q(a^*) = \max_{a \in \mathcal{A}} q_*(a)$$

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

$$I_t = \mathbb{E}[\nu_* - q_*(a_t)] \quad \text{reward} = -\text{regret}$$

- ▶ The **total regret** is the total opportunity loss

$$L_t = \mathbb{E} \left[ \sum_{t=1}^T \nu_* - q_*(a_t) \right]$$

- ▶ Maximize cumulative reward = minimize total regret

# Regret

- ▶ The **count**  $N_t(a)$ : the number of times that action  $a$  has been selected prior to time  $t$
- ▶ The **gap**  $\Delta_a$  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

$$\begin{aligned} L_t &= \mathbb{E} \left[ \sum_{t=1}^T v_* - q_*(a_t) \right] \\ &= \sum_{a \in \mathcal{A}} \mathbb{E}[N_t(a)](v_* - q_*(a)) \\ &= \sum_{a \in \mathcal{A}} \mathbb{E}[N_t(a)]\Delta_a \end{aligned}$$

# Forming Action-Value Estimates

- 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*

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



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

# Forming Action-Value Estimates

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

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

# Forming Action-Value Estimates

- 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:

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

# Forming Action-Value Estimates

- 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:

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

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

$$\text{NewEstimate} \leftarrow \text{OldEstimate} + \text{StepSize} [\text{Target} - \text{OldEstimate}]$$

# Forming Action-Value Estimates

- 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:

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

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

$$\text{NewEstimate} \leftarrow \text{OldEstimate} + \text{StepSize} [\text{Target} - \text{OldEstimate}]$$

# Derivation of incremental update

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

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

# Non-stationary bandits

- 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?

# Non-stationary bandits

- 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
- Better is an “exponential, recency-weighted average”:

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

where  $\alpha \in (0,1]$  and constant

The smaller the  $i$ , the smaller the multiplier-> forgetting earlier rewards

# This lecture

We have seen how to form estimates for the bandit mean rewards.  
Next we will discuss our action selection strategy (policy)

# Baseline: Fixed exploration period+Greedy

I. Allocate a fixed time period to exploration when you try bandits **uniformly at random**

# Baseline: Fixed exploration period+Greedy

1. Allocate a fixed time period to exploration when you try bandits **uniformly at random**

2. Estimate mean rewards for all actions

$$Q_t(a) = \frac{1}{N_t(a)} \sum_{i=1}^{t-1} r_i \mathbf{1}(A_i = a)$$

# Baseline: Fixed exploration period+Greedy

1. Allocate a fixed time period to exploration when you try bandits **uniformly at random**
2. Estimate mean rewards for all actions 
$$Q_t(a) = \frac{1}{N_t(a)} \sum_{i=1}^{t-1} r_i \mathbf{1}(A_i = a)$$
3. Select the action that is optimal for the estimated mean rewards given all data thus far, breaking ties at random

$$a_t = \operatorname{argmax}_{a \in \mathcal{A}} Q_t(a)$$

# Baseline: Fixed exploration period+Greedy

1. Allocate a fixed time period to exploration when you try bandits **uniformly at random**
2. Estimate mean rewards for all actions 
$$Q_t(a) = \frac{1}{N_t(a)} \sum_{i=1}^{t-1} r_i \mathbf{1}(A_i = a)$$
3. Select the action that is optimal for the estimated mean rewards given all data thus far, breaking ties at random  
$$a_t = \operatorname{argmax}_{a \in \mathcal{A}} Q_t(a)$$
4. GOTO 3

# Baseline: Fixed exploration period + Greedy

After the fixed exploration period we have formed the following reward estimates



$$Q_t(a_1) = 0.3$$

$$Q_t(a_2) = 0.5$$

$$Q_t(a_3) = 0.1$$

Q: Will the greedy method always pick the second action?

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

# $\epsilon$ -Greedy Action Selection

- In greedy action selection, you always exploit
- In  $\epsilon$ -greedy, you are usually greedy, but with probability  $\epsilon$  you instead pick an action at random (possibly the greedy action again)
- This is perhaps the simplest way to balance exploration and exploitation

# $\varepsilon$ -Greedy Action Selection

## A simple bandit algorithm

Initialize, for  $a = 1$  to  $k$ :

$$\begin{aligned} Q(a) &\leftarrow 0 \\ N(a) &\leftarrow 0 \end{aligned}$$

Repeat forever:

$$\begin{aligned} A &\leftarrow \begin{cases} \arg \max_a Q(a) & \text{with probability } 1 - \varepsilon \\ \text{a random action} & \text{with probability } \varepsilon \end{cases} \quad (\text{breaking ties randomly}) \\ R &\leftarrow \text{bandit}(A) \\ N(A) &\leftarrow N(A) + 1 \\ Q(A) &\leftarrow Q(A) + \frac{1}{N(A)} [R - Q(A)] \end{aligned}$$

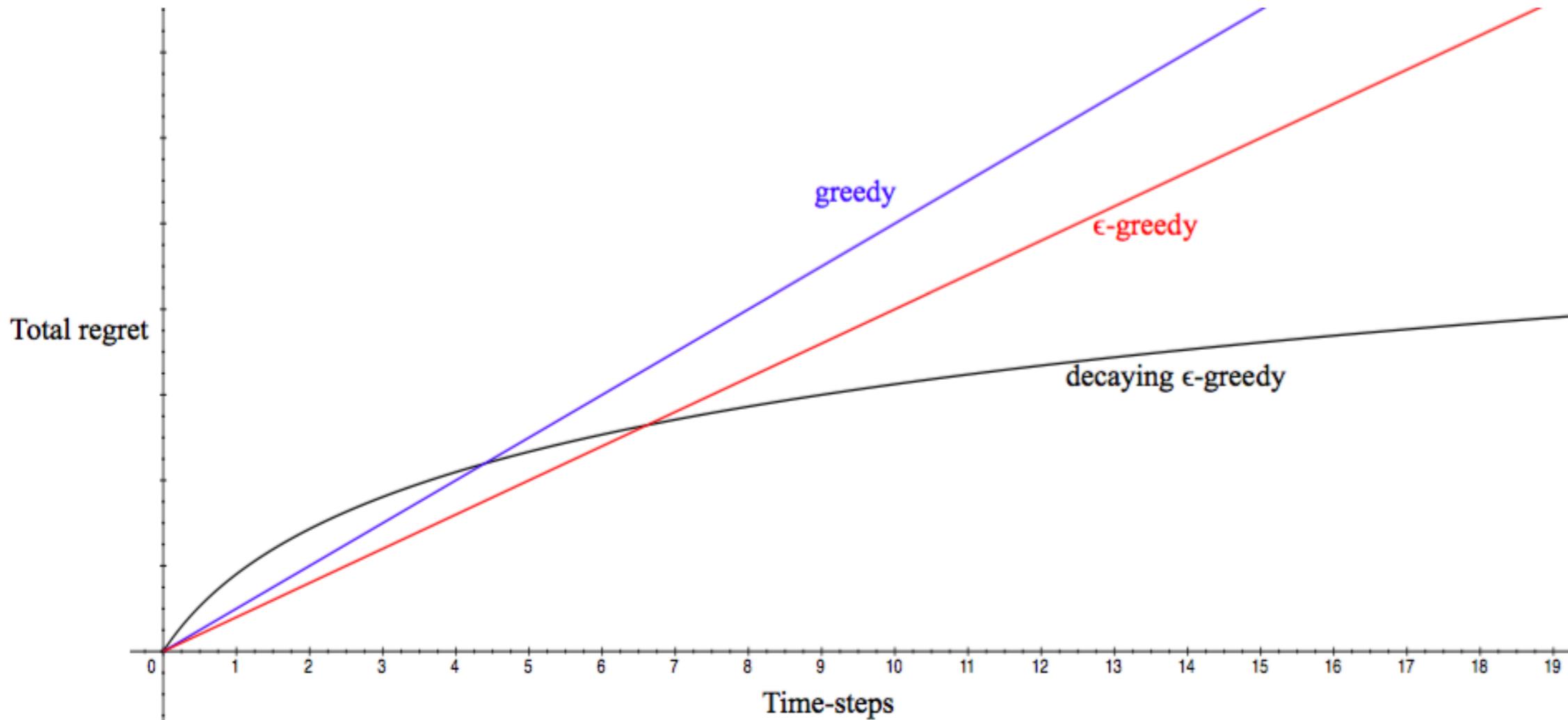
# $\epsilon$ -Greedy Algorithm

- ▶ The  $\epsilon$ -greedy algorithm continues to explore forever
  - With probability  $1 - \epsilon$  select  $a_t = \operatorname{argmax}_{a \in \mathcal{A}} Q_t(a)$
  - With probability  $\epsilon$  select a random action
- ▶ Constant  $\epsilon$  ensures minimum regret

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

- ▶  $\Rightarrow \epsilon$ -greedy has linear total regret

# Counting Regret

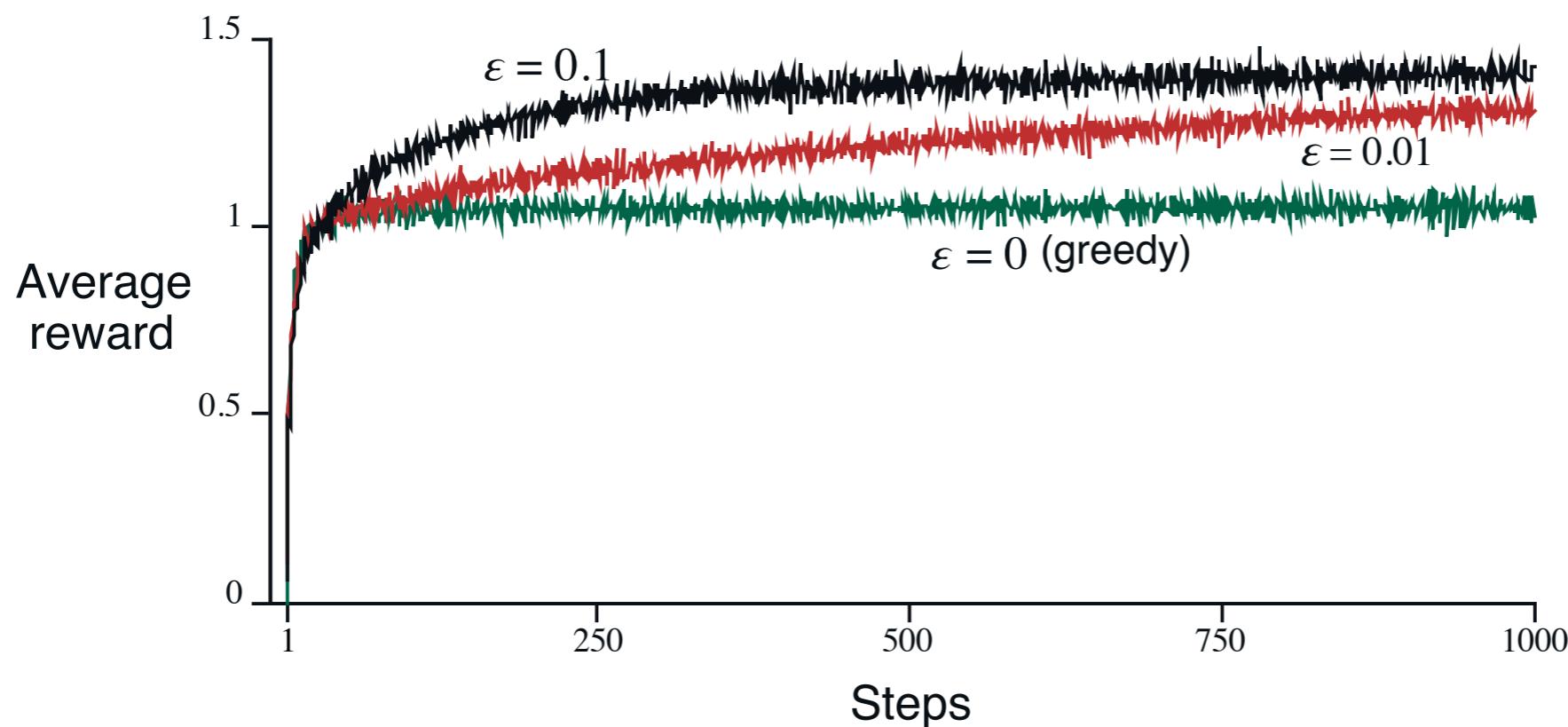
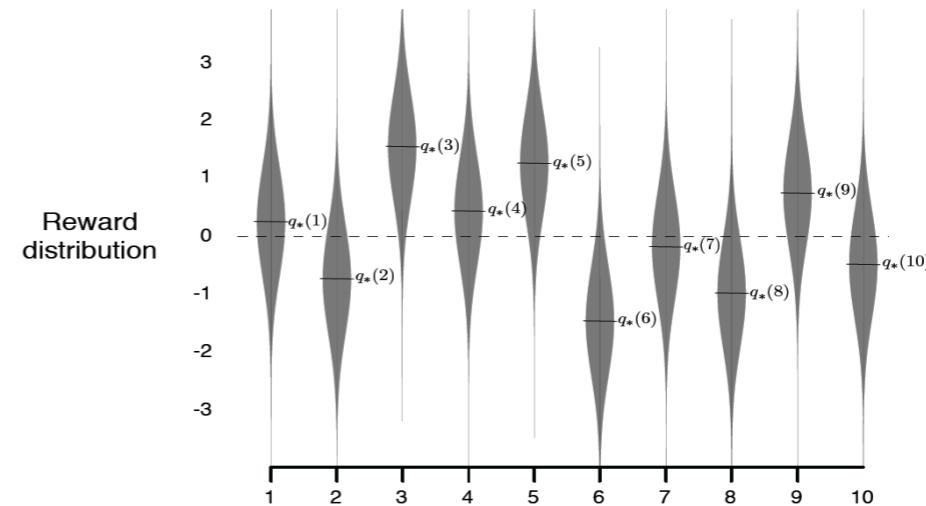


- ▶ If an algorithm forever explores it will have linear total regret
- ▶ If an algorithm never explores it will have linear total regret

# Average reward for three algorithms

We sample 10 arm bandits instantiations:

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

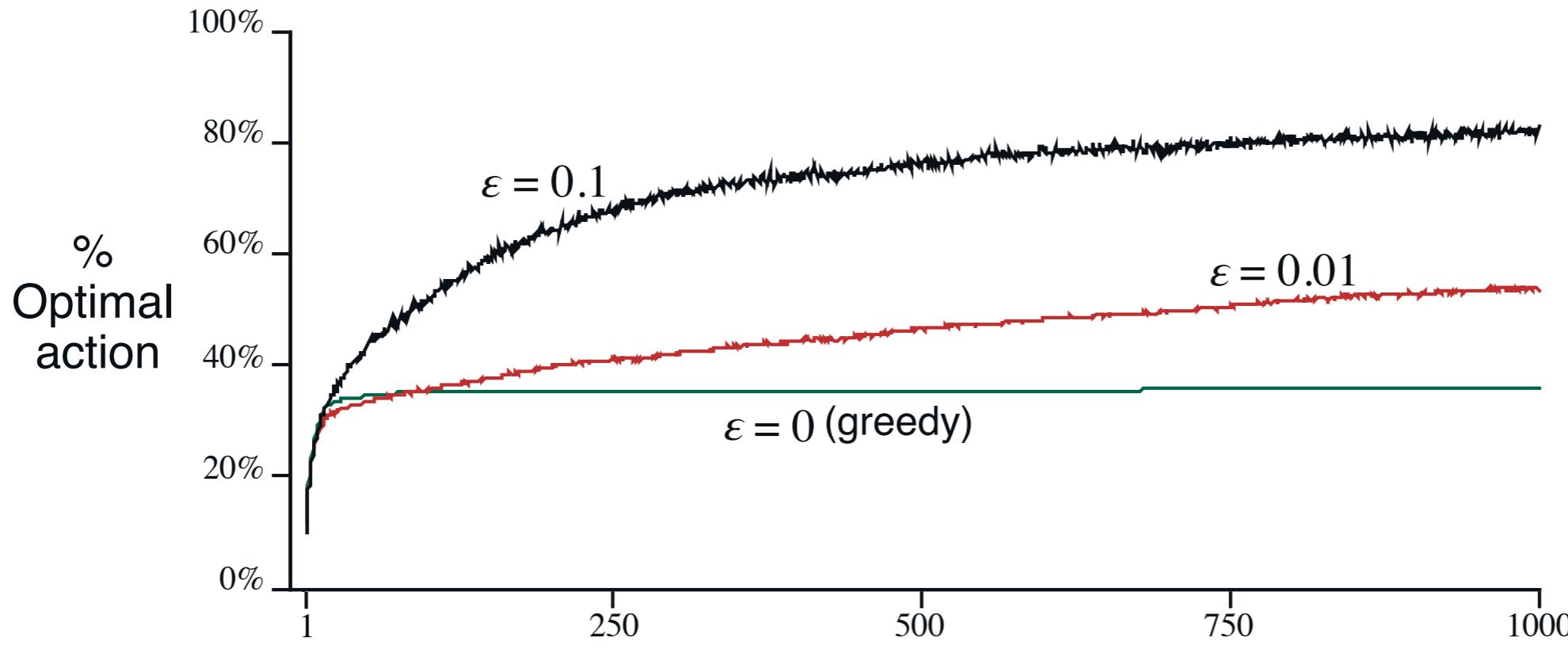
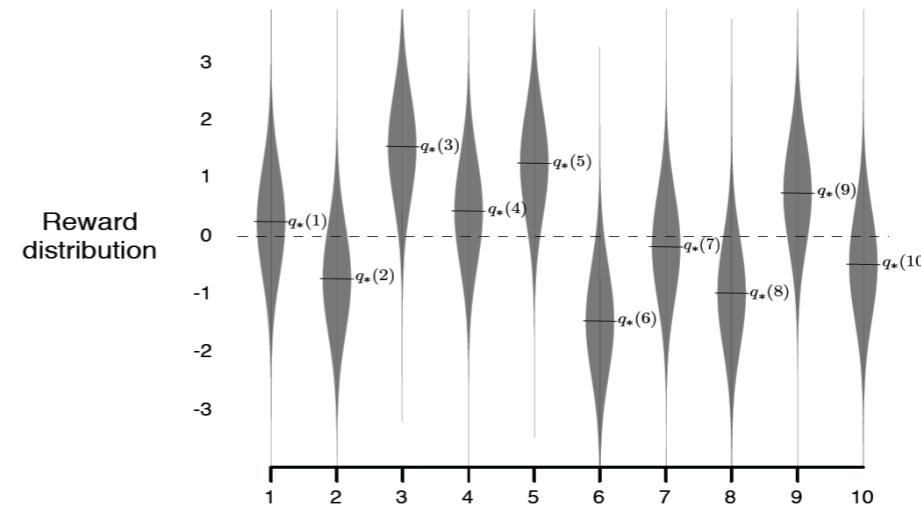


Q: In the limit (after infinite number of steps), which method will result in the largest average reward?

# Optimal action for three algorithms

We sample 10 arm bandits instantiations from here

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

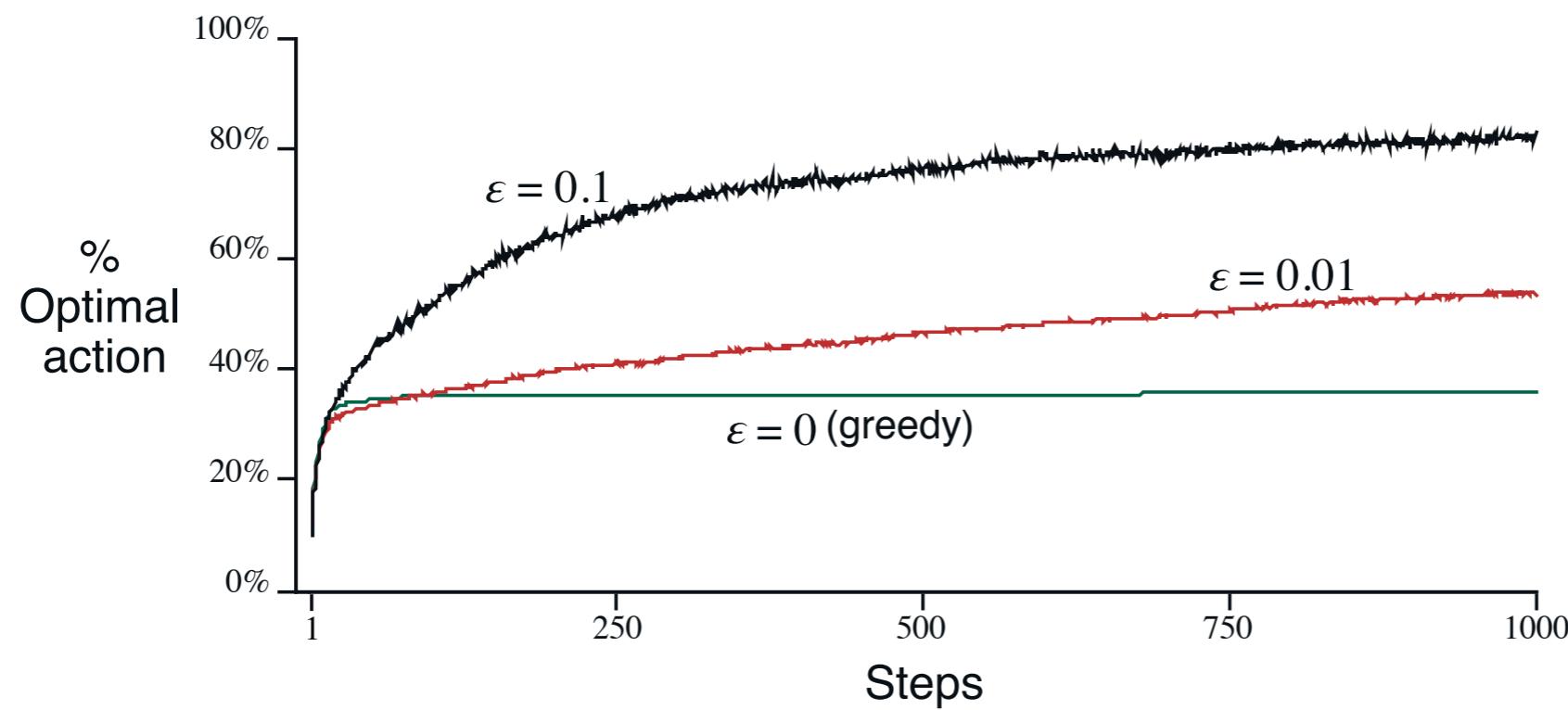
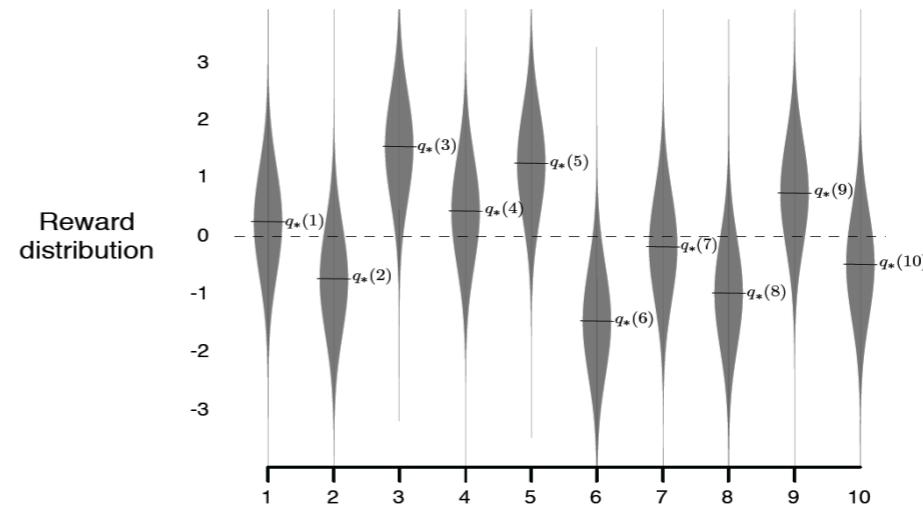


Q: Which method will find the optimal action in the limit?

# Optimal action for three algorithms

We sample 10 arm bandits instantiations from here

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



Q: Does the performance of those methods depend on the initialization of the action value estimates?

# Optimistic Initialization

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

$$Q_t(a_t) = Q_{t-1}(a_t) + \frac{1}{N_t(a_t)}(r_t - Q_{t-1}(a_t))$$

- ▶ 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

# Optimistic Initial Values

We initialize with the following reward estimates for Bernoulli bandits



$$Q_t(a_1) = 1$$



$$Q_t(a_2) = 1$$

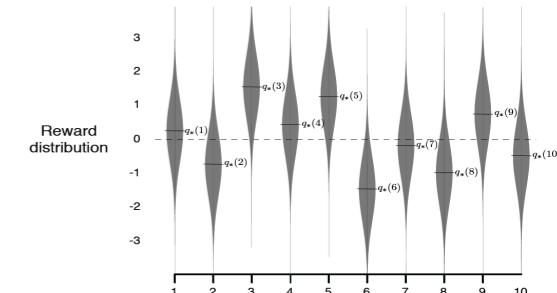


$$Q_t(a_3) = 1$$

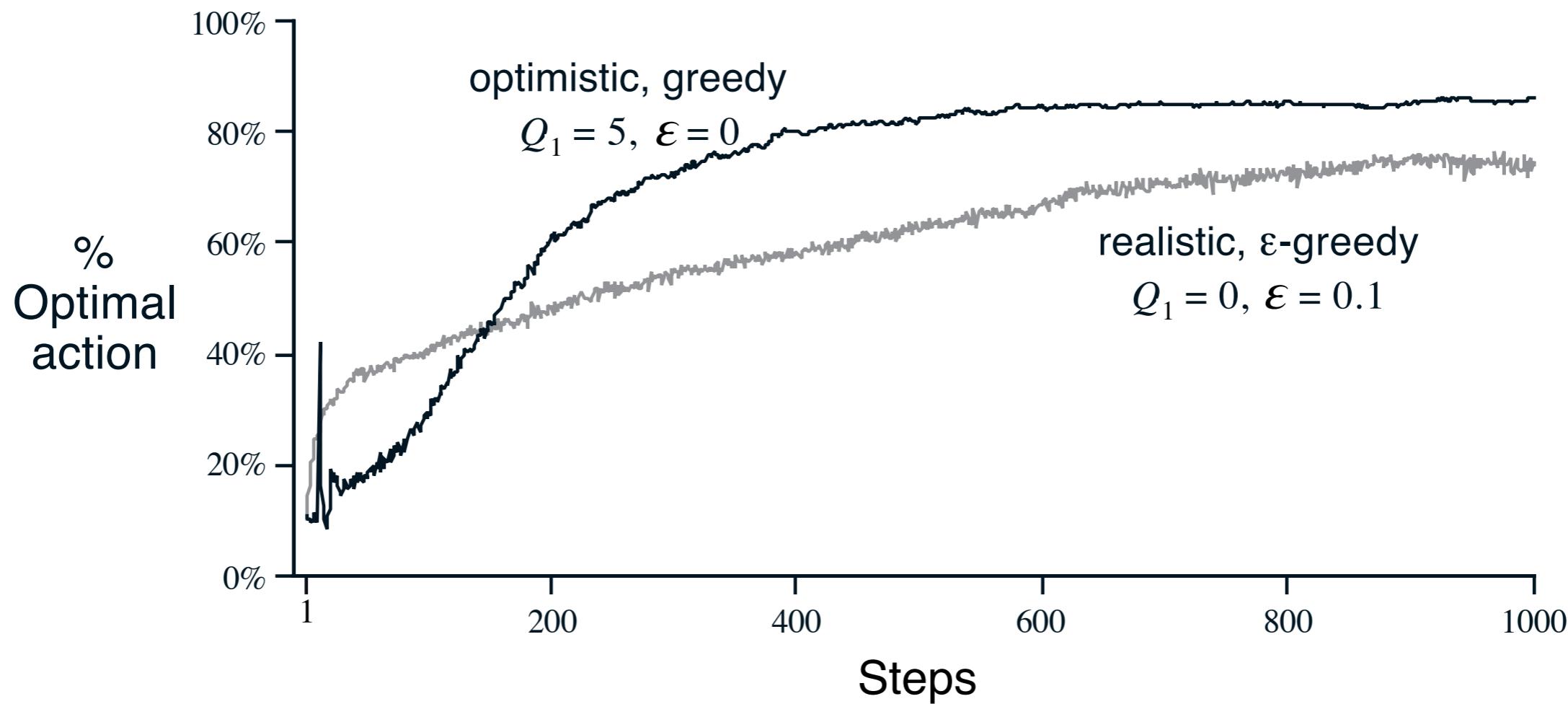
Q: When it is possible that greedy action selection will not try out all the actions?

# Optimistic Initial Values

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



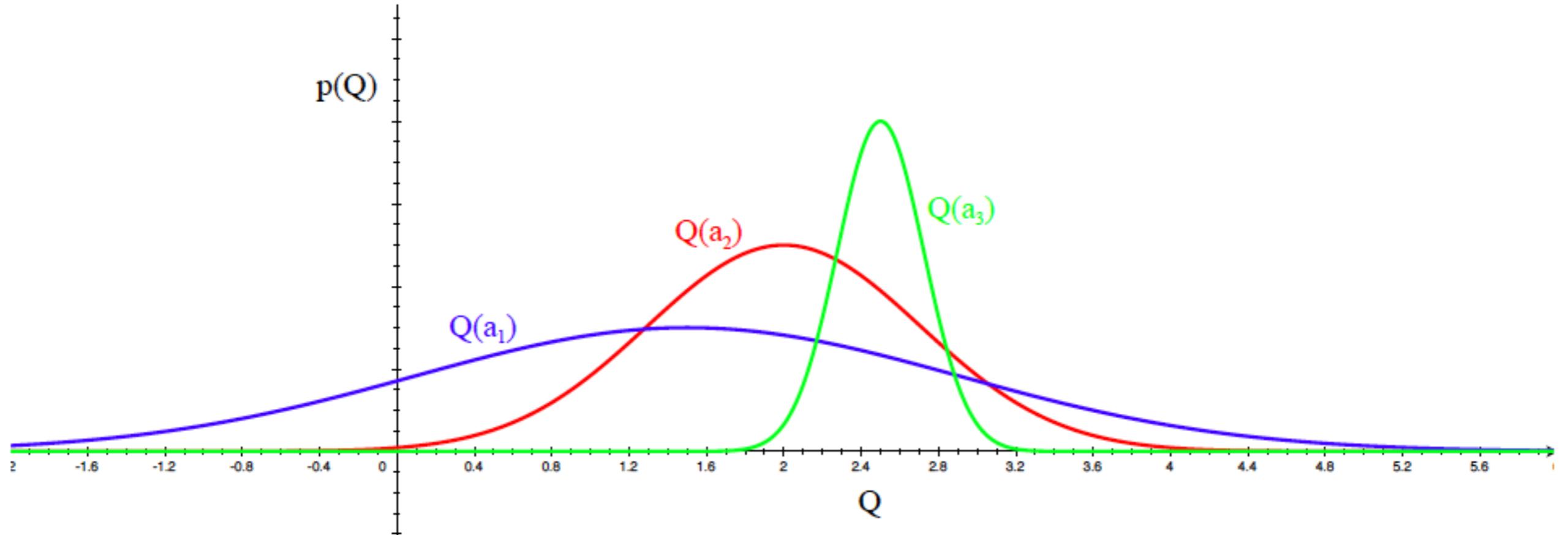
- Suppose we initialize the action values *optimistically* ( $Q_1(a) = 5$ ), e.g., on the 10-armed testbed



- **Goal: find an algorithm with sub-linear regret for any multi-armed bandit**

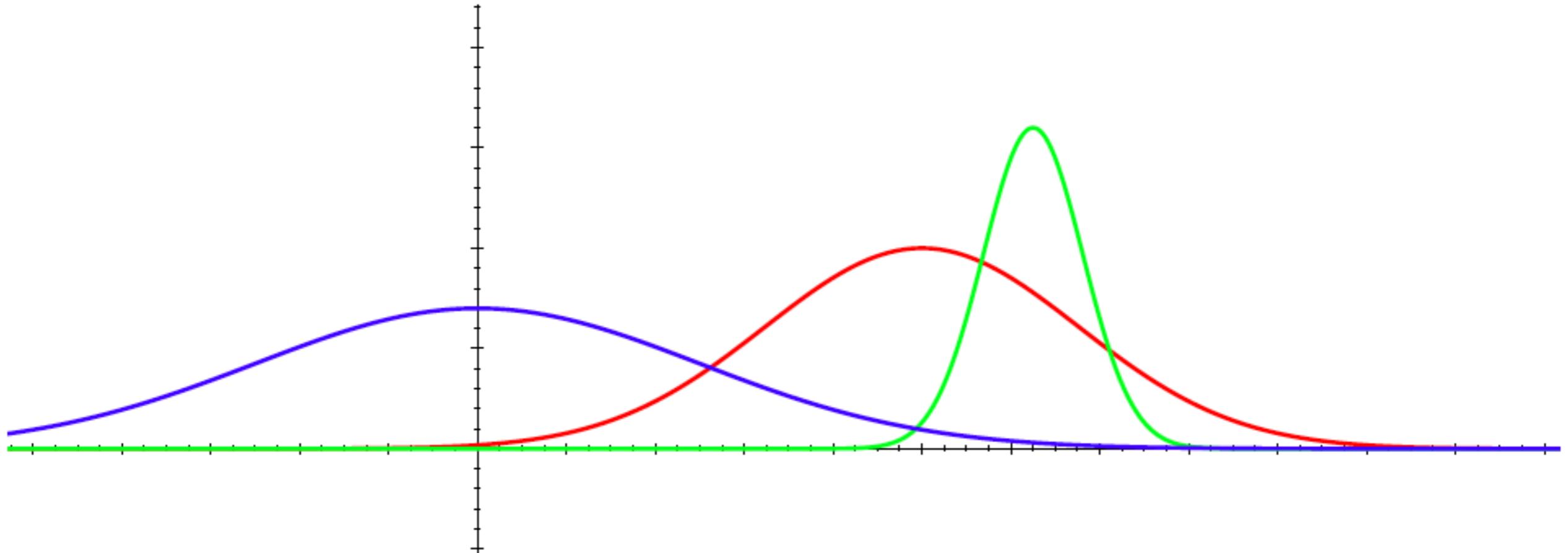
To achieve that we need to reason about **uncertainty** of our action value estimates

# 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

$$q_*(a) \leq Q_t(a) + U_t(a)$$



- ▶ This upper confidence depends on the number of times action  $a$  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)**

$$a_t = \operatorname{argmax}_{a \in \mathcal{A}} Q_t(a) + U_t(a)$$

# 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  $a$

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

# Calculating Upper Confidence Bounds

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

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

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

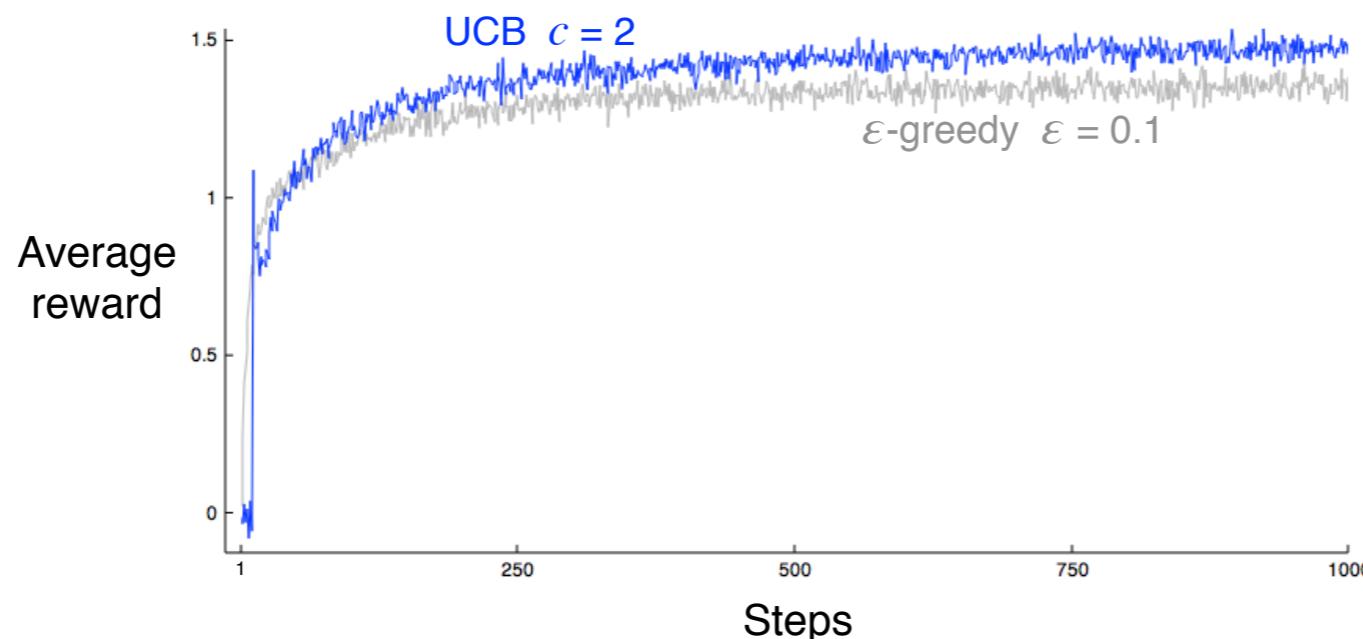
- ▶ Reduce  $p$  as we observe more rewards, e.g.  $p = t^{-c}$ ,  $c=4$   
(note:  $c$  is a hyper-parameter that trades-off explore/exploit)
- ▶ Ensures we select optimal action as  $t \rightarrow \infty$

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

# Upper Confidence Bound (UCB)

- A clever way of reducing exploration over time
- Estimate an upper bound on the true action values
- Select the action with the largest (estimated) upper bound

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





1000 pulls,  
600 wins  
 $Q_t(a_1)=0.6$



1000 pulls,  
400 wins  
 $Q_t(a_2)=0.4$



10 pulls,  
4 wins  
 $Q_t(a_1)=0.4$

## Epsilon-greedy

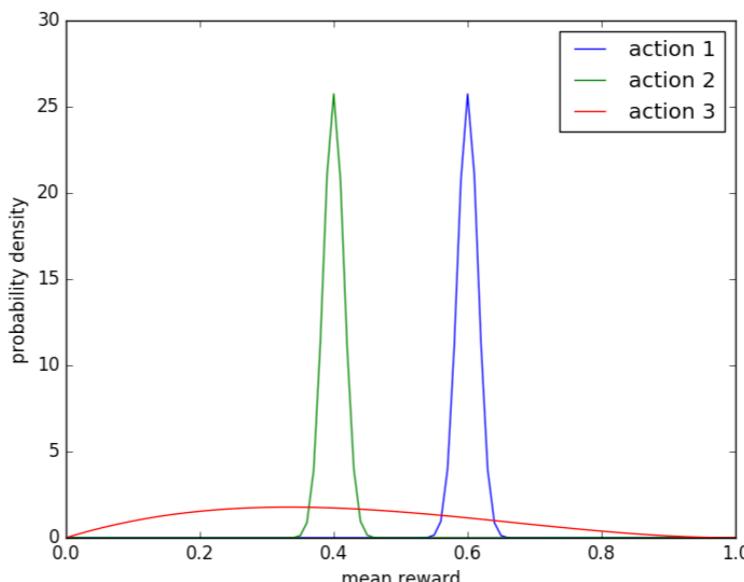
Repeat forever:

```

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

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

The problem with using mean estimates is that we cannot reason about uncertainty of those estimates..



# 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:  $h_t = a_1, r_1, \dots, a_{t-1}, r_{t-1}$
- ▶ Use posterior to guide exploration

# Bayes rule

*Bayes rule enables us to reverse probabilities:*

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$



# Problem 1: Diagnoses

- The doctor has bad news and good news.
- The bad news is that you tested positive for a serious disease, and that **the test is 99% accurate** (i.e., the probability of testing positive given that you have the disease is 0.99, as is the probability of testing negative given that you don't have the disease).
- The good news is that this is a rare disease, striking only 1 in 10,000 people.
- What are the chances that you actually have the disease?

# Problem 1: Diagnoses

The test is 99% accurate:  $P(T=1|D=1) = 0.99$  and  $P(T=0|D=0) = 0.99$

Where  $T$  denotes test and  $D$  denotes disease.

The disease affects 1 in 10000:  $P(D=1) = 0.0001$

# Problem 1: Diagnoses

The test is 99% accurate:  $P(T=1|D=1) = 0.99$  and  $P(T=0|D=0) = 0.99$   
Where  $T$  denotes test and  $D$  denotes disease.

The disease affects 1 in 10000:  $P(D=1) = 0.0001$

$$P(D=1 | T=1) = \frac{P(T=1 | D=1) P(D=1)}{P(T=1 | D=0) P(D=0) + P(T=1 | D=1) P(D=1)}$$

# Problem 1: Diagnoses

The test is 99% accurate:  $P(T=1|D=1) = 0.99$  and  $P(T=0|D=0) = 0.99$   
Where  $T$  denotes test and  $D$  denotes disease.

The disease affects 1 in 10000:  $P(D=1) = 0.0001$

$$P(D=1 | T=1) = \frac{P(T=1 | D=1) P(D=1)}{P(T=1 | D=0) P(D=0) + P(T=1 | D=1) P(D=1)}$$
$$\approx 0.0098$$

# Bayesian learning for model parameters

**Step 1:** Given  $\textcolor{violet}{n}$  data,  $\mathbf{D} = \mathbf{x}_{1:\textcolor{violet}{n}} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{\textcolor{violet}{n}}\}$ , write down the expression for the *likelihood*:

$$p(\mathbf{D} | \theta)$$

**Step 2:** Specify a *prior*:  $p(\theta)$

**Step 3:** Compute the *posterior*:

$$p(\theta | \mathbf{D}) = \frac{p(\mathbf{D} | \theta) p(\theta)}{p(\mathbf{D})}$$

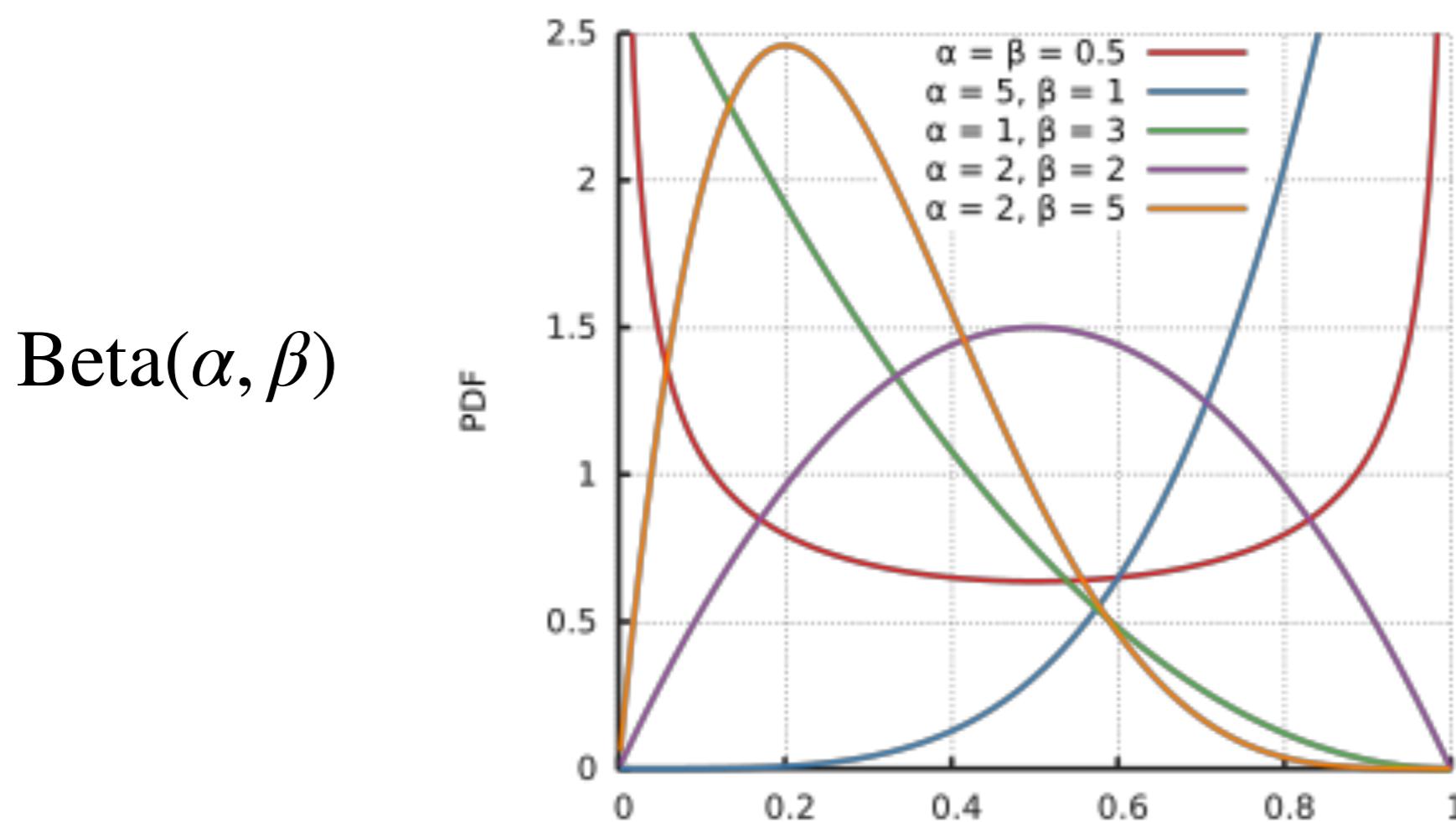
# Bernoulli bandits - Prior

Let's consider a Beta distribution prior over the mean rewards of the Bernoulli bandits:

$$p(\theta_k) = \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \theta_k^{\alpha_k-1} (1-\theta_k)^{\beta_k-1} \quad \Gamma(n) = (n-1)!$$

The mean is  $\frac{\alpha}{\alpha + \beta}$

The larger the  $\alpha + \beta$  the more concentrated the distribution



# Bernoulli bandits-Posterior

Let's consider a Beta distribution **prior** over the mean rewards of the Bernoulli bandits:

$$p(\theta_k) = \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \theta_k^{\alpha_k-1} (1 - \theta_k)^{\beta_k-1}$$
$$\Gamma(n) = (n - 1)!$$

$$p(\theta | D) = \frac{p(D | \theta) p(\theta)}{p(D)}$$

The posterior is also a Beta! Because beta is conjugate distribution for the Bernoulli distribution.

A closed form solution for the bayesian update, possible only for conjugate distributions!

$$(\alpha_k, \beta_k) \leftarrow \begin{cases} (\alpha_k, \beta_k) & \text{if } x_t \neq k \\ (\alpha_k, \beta_k) + (r_t, 1 - r_t) & \text{if } x_t = k. \end{cases}$$

# Greedy VS Thompson for Bernoulli bandits

---

**Algorithm 1** BernGreedy( $K, \alpha, \beta$ )

---

```
1: for  $t = 1, 2, \dots$  do
2:   #estimate model:
3:   for  $k = 1, \dots, K$  do
4:      $\hat{\theta}_k \leftarrow \alpha_k / (\alpha_k + \beta_k)$ 
5:   end for
6:
7:   #select and apply action:
8:    $x_t \leftarrow \operatorname{argmax}_k \hat{\theta}_k$ 
9:   Apply  $x_t$  and observe  $r_t$ 
10:
11:  #update distribution:
12:   $(\alpha_{x_t}, \beta_{x_t}) \leftarrow (\alpha_{x_t}, \beta_{x_t}) + (r_t, 1 - r_t)$ 
13: end for
```

---

a: success  
b: failure

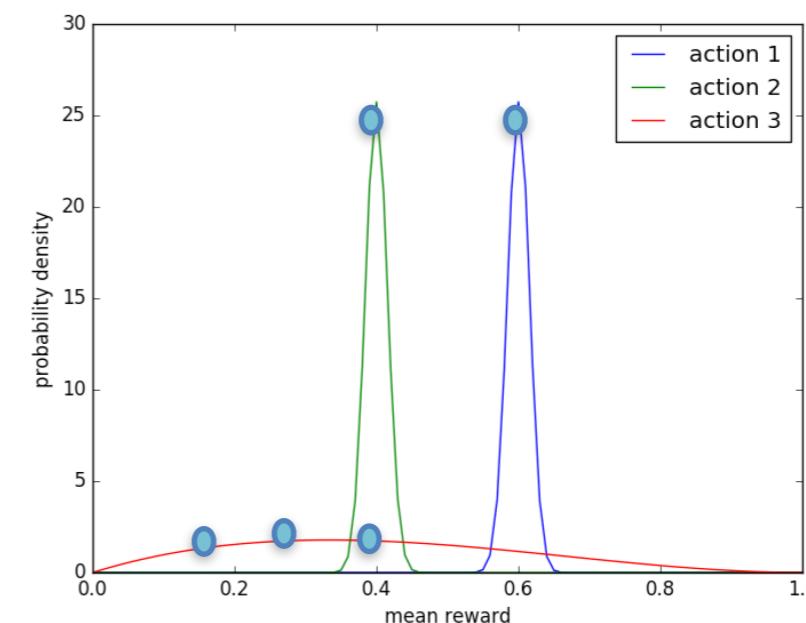
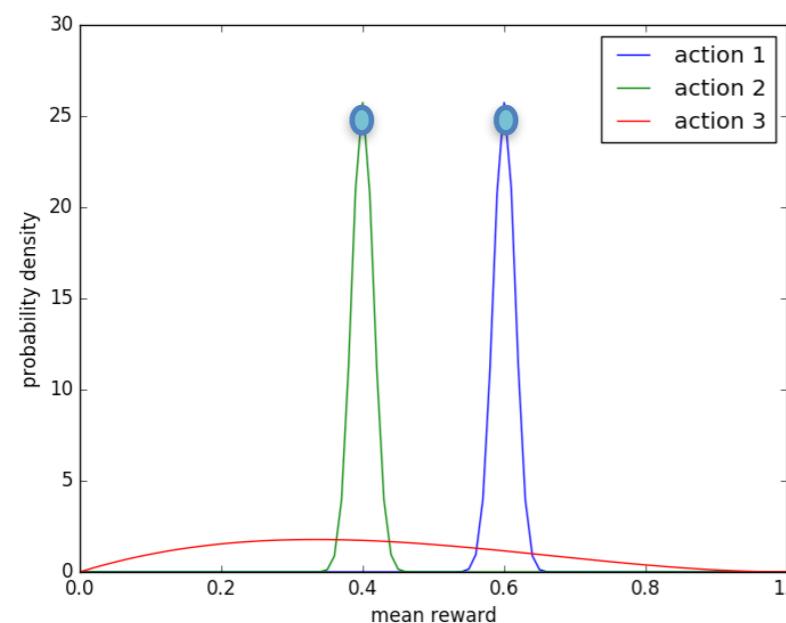
---

**Algorithm 2** BernThompson( $K, \alpha, \beta$ )

---

```
1: for  $t = 1, 2, \dots$  do
2:   #sample model:
3:   for  $k = 1, \dots, K$  do
4:     Sample  $\hat{\theta}_k \sim \text{beta}(\alpha_k, \beta_k)$ 
5:   end for
6:
7:   #select and apply action:
8:    $x_t \leftarrow \operatorname{argmax}_k \hat{\theta}_k$ 
9:   Apply  $x_t$  and observe  $r_t$ 
10:
11:  #update distribution:
12:   $(\alpha_{x_t}, \beta_{x_t}) \leftarrow (\alpha_{x_t}, \beta_{x_t}) + (r_t, 1 - r_t)$ 
13: end for
```

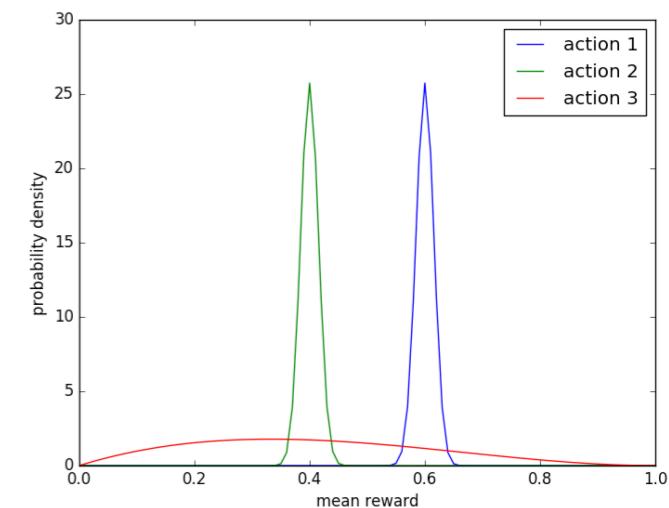
---



# Recall: Thompson Sampling

Represent a posterior distribution of mean rewards of the bandits, as opposed to mean estimates.

1. Sample from it  $\theta_1, \theta_2, \dots, \theta_k \sim \hat{p}(\theta_1, \theta_2 \dots \theta_k)$
2. Choose action  $a = \arg \max_a \mathbb{E}_\theta[r(a)]$
3. Update the mean reward distribution  $\hat{p}(\theta_1, \theta_2 \dots \theta_k)$



The equivalent of mean expected rewards for general MDPs are Q functions

# 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  $\sum_{\tau=1}^t r_\tau$



# Real world motivation: Personalized NETFLIX artwork



For a particular title **and a particular user**, we will use the **contextual** multi-armed bandit formulation to decide what image to show per title **per user**

- Actions: uploading an image (available for this movie title) to a user's home screen
- Mean rewards (unknown): the % of NETFLIX users that will click on the title and watch the movie
- Estimated mean rewards: the average click rate (+quality engagement, not clickbait)
- **Context** (s) : user attributes, e.g., language preferences, gender of films she has watched, time and day of the week, etc.