Reinforcement Learning

The Multi-arm Bandit Problem

Debapriyo Majumdar

Indian Statistical Institute

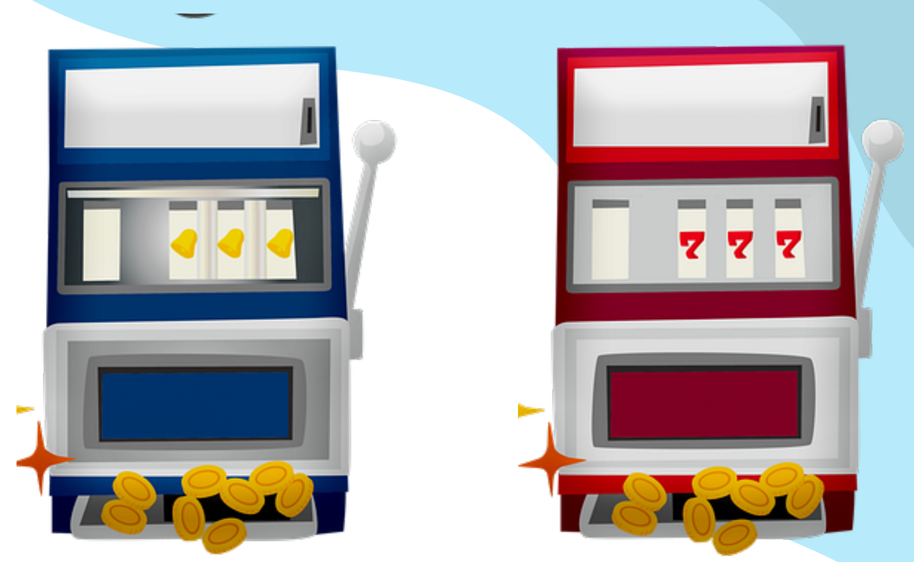
debapriyo@isical.ac.in

The k-arm bandit problem

- There are k slot machines (or one with k arms), each with a **stationary probability distribution** of rewards
  - The probability distributions are not known to us
- At each step t, we take an **action** A_t, i.e., choose to draw from one of the k slot machines
  - We get a **reward** $R(t)$
- Goal: maximize total reward over a time period (say N actions)
- Theme:
  - Know nothing to start with
  - As we keep getting rewards (high or low), we learn how to take better decisions in future

The k-arm bandit problem

……



- There are k slot machines, each with a **stationary probability distribution** of rewards
  - Each action has a mean reward, called the **value** of the action
  - Value of an arbitrary action a, called $q_*(a) = \mathbb{E}(R_t \mid A_t = a)$
- However, we don't know $q_*(a)$, if we knew we could always choose the one with highest value
  - We can only estimate it
- The estimate keeps changing with time t as well
- Define $Q_t(a)$ = estimated value of action a at step t
- Goal: try to make $Q_t(a)$ as close to $q_*(a)$ as possible

Exploitation and exploration

- Exploitation (greedy approach): at each step t, pick the action a for which the estimate $Q_t(a)$ is highest
  - Maximizes reward for the next step
- Exploration: pick some non-greedy action
  - May help improving estimates better
  - May maximize reward in the long run
- Balance exploitation and exploration
- We will discuss simple techniques which do not depend on unrealistic assumptions

Action - value methods

- How to estimate the value of an action?
- Sample average method: estimate the value of an action a by averaging the rewards actually received from action a
- Define: the indicator function $\mathbf{1}_P$ for any predicate P as 0 if P is false and 1 if P is true
- Then

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

- The value is some default (say 0) if action a has not been taken yet (denominator is 0)
- Now, having the estimate, which action to choose?

Greedy method

- Select the action with the highest estimated reward at that point
  - Greedy: $A_t = \arg\max_a Q_t(a)$
  - Problem: not all actions may be even taken ever
- Alternative to completely greedy method: ε-greedy
  - With probability $(1 - \varepsilon)$ take the greedy action
  - With probability ε choose some action randomly from all available options
  - Advantage: eventually all actions will be taken "many" times
    - For every action a, $Q_t(a)$ will converge to $q_*(a)$

A simple bandit algorithm

- If the average of a sequence $R_1, R_2, \ldots, R_{n-1}$ is Q_n, then the average of the sequence $R_1, R_2, \ldots, R_{n-1}, R_n$ can be computed as: $Q_{n+1} = Q_n + \frac{1}{n}(R_n - Q_n)$
- **Simple bandit algorithm (ε-greedy)**

Initialize, for $a = 1,\ldots,k$:

Estimate for action $Q(a) = 0$, number of times action a is taken $N(a) = 0$

while (true):

Draw a random number $x \in (0,1)$

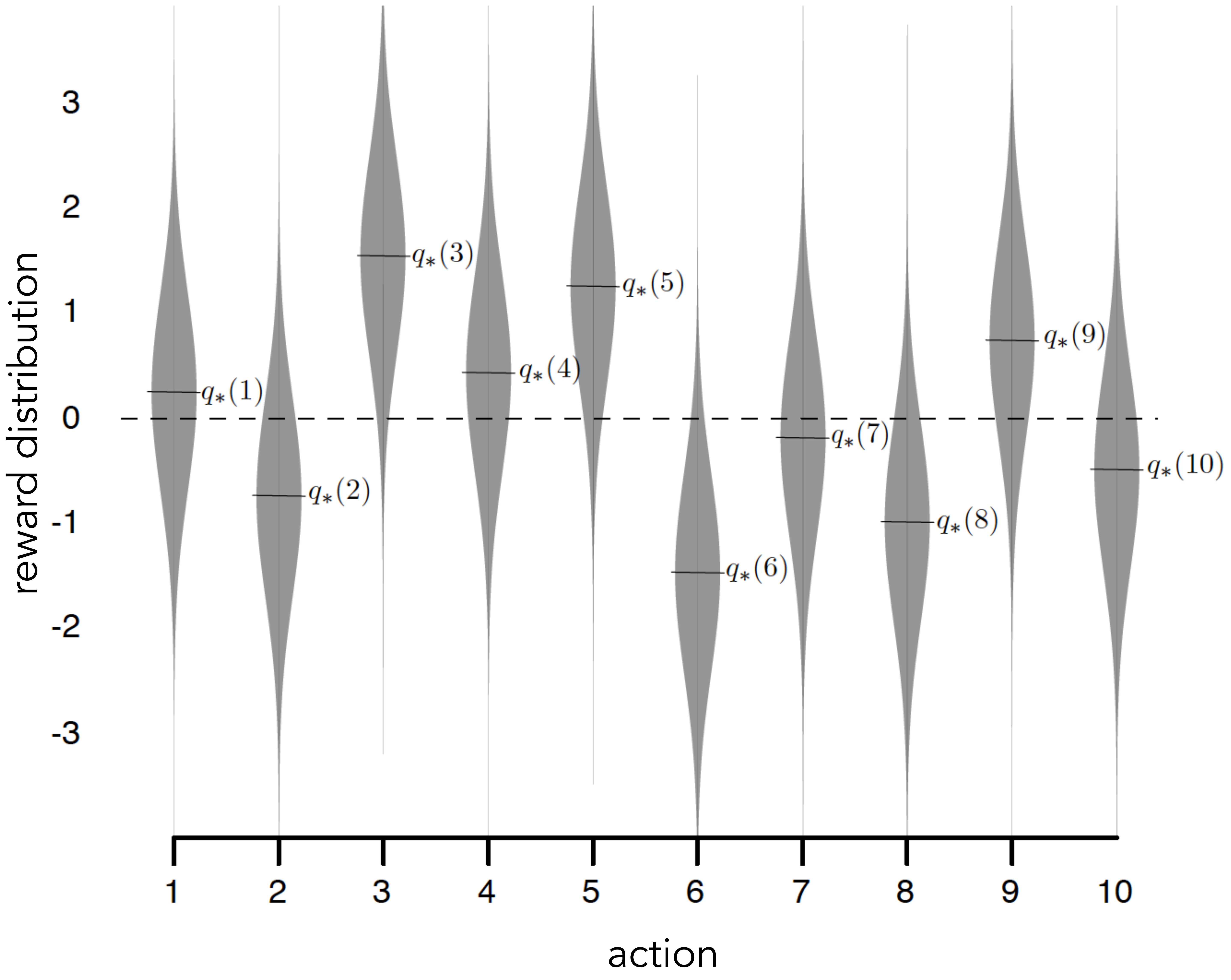
if $x < \varepsilon$, $A =$ a random action, else $A = \underset{a}{\arg\max}\ Q(a)$

Current reward $R = \text{bandit}(A)$ [reward obtained from action A at this instance]

Increment count $N(A) = N(A) + 1$

Update estimate $Q(A) = Q(A) + \frac{1}{N(A)}(R - Q(A))$

Experiment: k-armed testbed ($k = 10$)

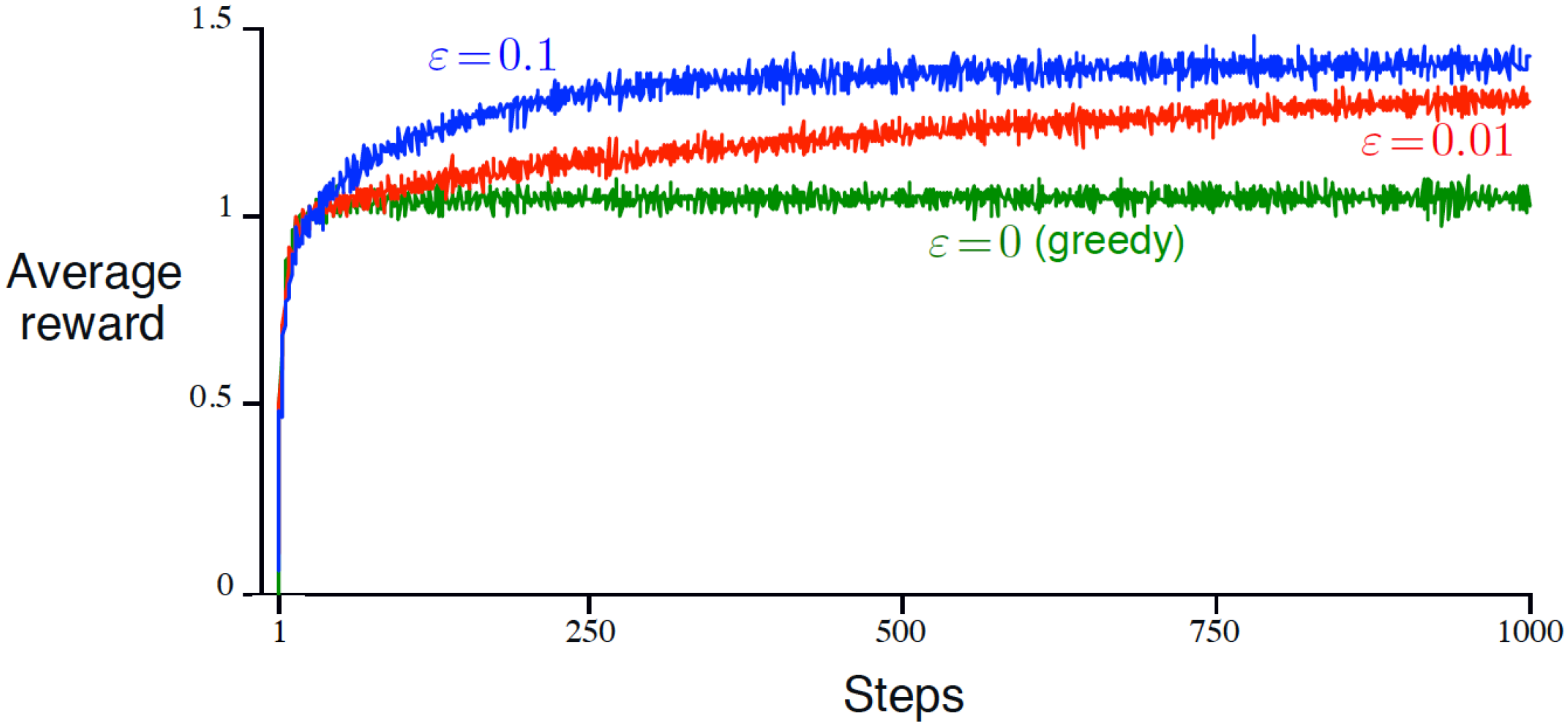


Picture courtesy: Sutton and Barto, 2018

- Generate k random numbers from $\mathcal{N}(0,1)$ as $q_*(a)$ for $a = 1,2,\ldots,k$
- Consider a k-armed bandit where each action a has reward distribution following mean $q_*(a)$ and variance 1, i.e., following $\mathcal{N}(q_*(a),1)$
- Simulate different algorithms on this testbed
- One particular simulation is not reliable, so consider average performance over many (say 2000) bandits generated this way

Greedy vs ε-greedy

- Greedy performs better initially
- However, ε-greedy methods performs more explorations and eventually outperforms greedy
- Eventually, $\varepsilon = 0.01$ would outperform the other two (why?)



Picture courtesy: Sutton and Barto, 2018

General form of the update rule

New estimate = Old estimate + Step size (Target – Old estimate)

$$Q(a) = Q(a) + \alpha_t(a)[R_t(a) - Q(a)]$$

For the algorithm we have seen, the step size

$$\alpha_t(a) = \frac{1}{N(a)}.$$

Non-stationary bandit problem

- Non-stationary probability: the reward probability distribution may change over time
  - More towards real world
- Need to have higher weightage for more recent rewards
- One possible approach: **exponential recency-weighted average**
  - Can be achieved by a constant step size (instead of a decaying step size) $\alpha_t(a) = \alpha$
- Then:

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

- Using this repeatedly by backward induction, we obtain

$$Q_{n+1} = (1-\alpha)^n Q_1 + \alpha(1-\alpha)^{n-1}R_1 + \alpha(1-\alpha)^{n-2}R_2 + \ldots + \alpha(1-\alpha)R_{n-1} + \alpha R_n$$

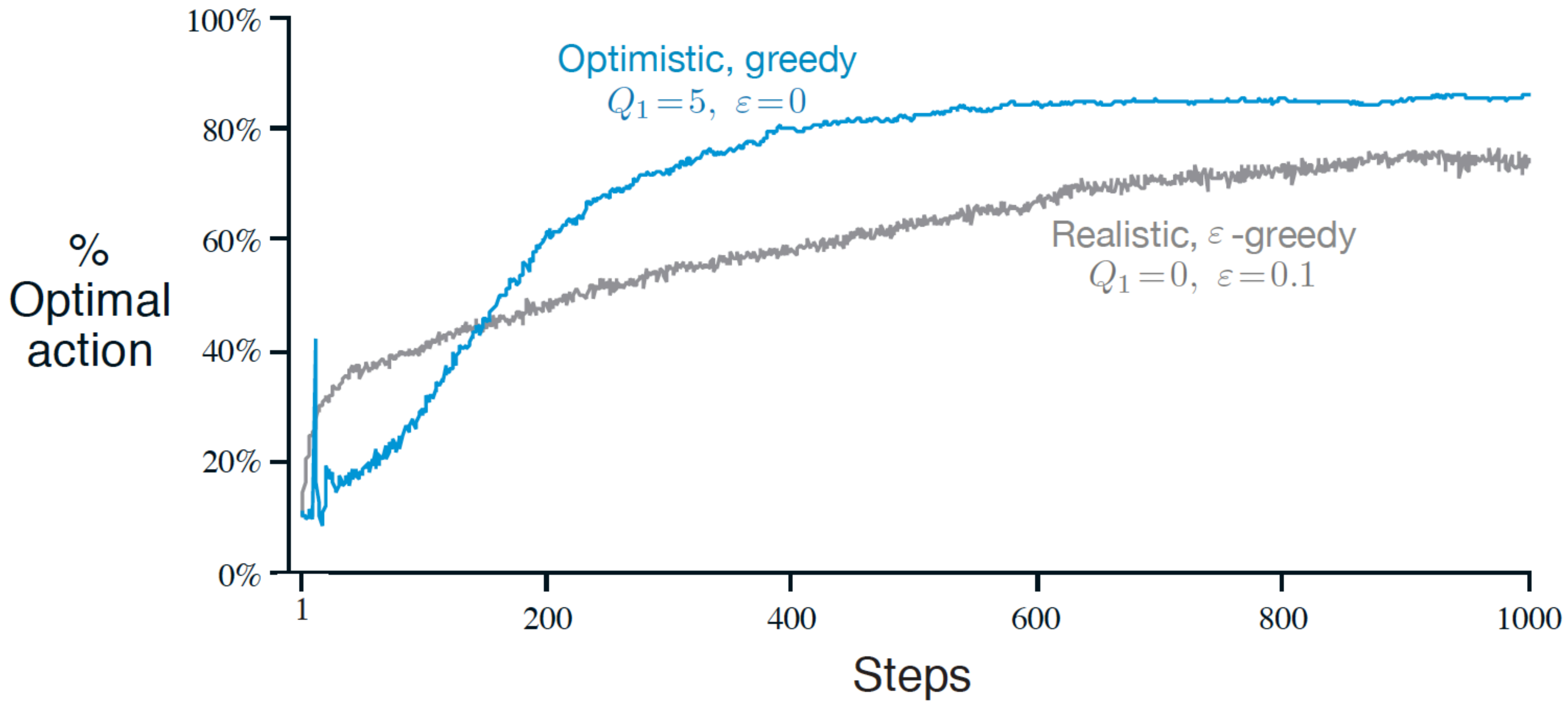
- Weightage for older Q_i decreases exponentially

Convergence of estimates to the actual value

- Goal: We want $Q_n(a) \rightarrow q_*(a)$ for large n
- Result: using the update rule $Q(a) = Q(a) + \alpha_t(a)[R_t(a) - Q(a)]$, the estimate converges with probability 1 if,
  - The steps are large enough to overcome initial effect of random fluctuations: $\sum_{t=1}^{\infty} \alpha_t(a) = \infty$, and
  - The steps are small enough to ensure convergence: $\sum_{t=1}^{\infty} \alpha_t^2(a) < \infty$
- For step size $\alpha_t(a) = \frac{1}{t}$, the conditions are satisfied, the estimates converge
- For step size $\alpha_t(a) = \alpha$ (constant), the second condition is not satisfied
  - The estimates continue to vary with recent fluctuations
  - But, that is desirable in a non-stationary environment

Optimistic initial value

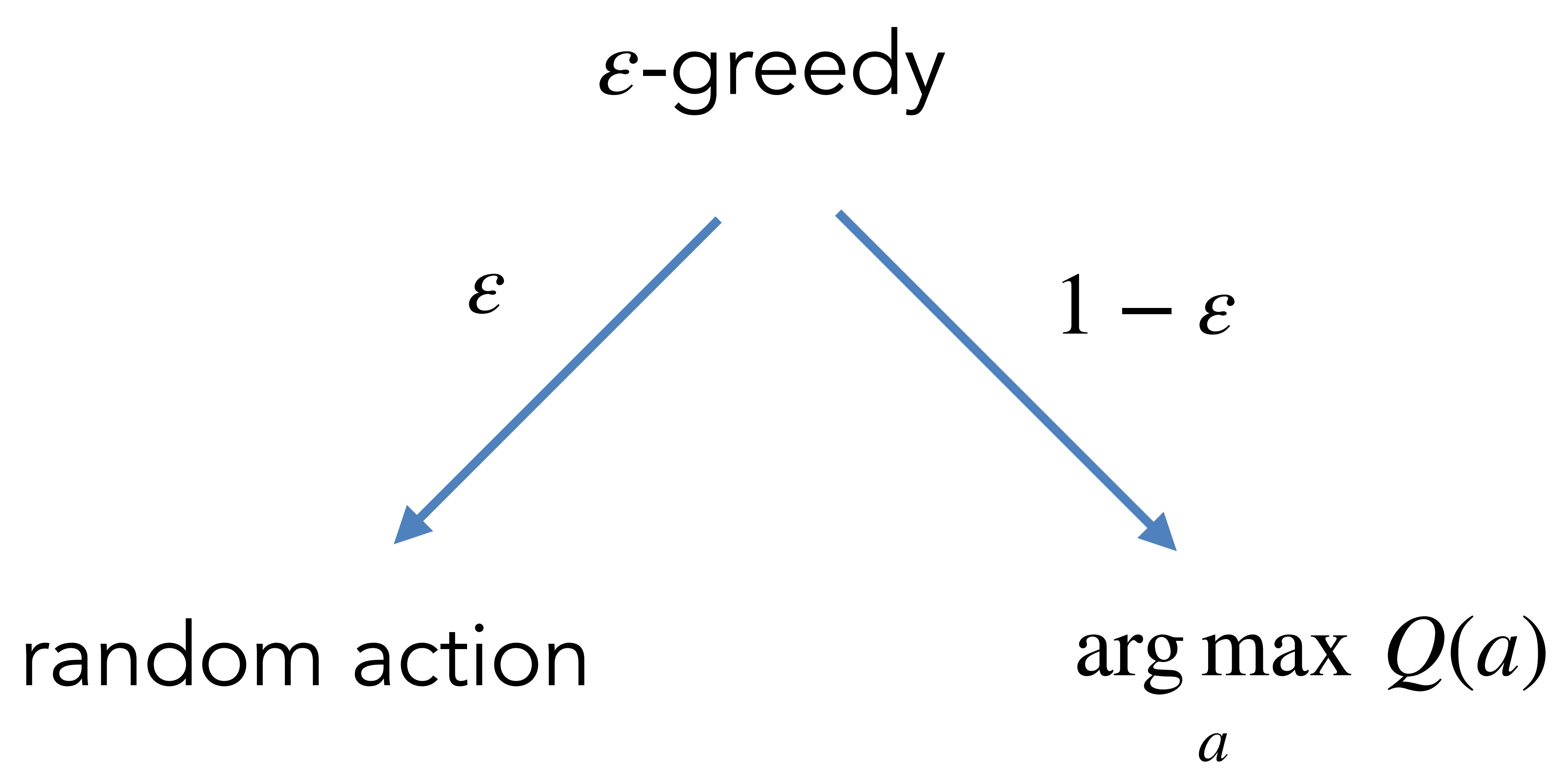
- Another approach for *encouraging* exploration: have high initial values for rewards
- Zero initialization: all estimates start with zero
- Optimistic initialization: initialize all estimates to some number much higher than $q_*(a)$ for all a
  - Time $t = 1$: some action will be chosen, but the value will be < the estimate (high initialization)
  - Disappointed: will choose some other action next time (but will be disappointed again)
  - During the initial stage, will ensure all actions are explored (but that's temporary)



Average performance of optimistic greedy vs realistic ε-greedy

Picture courtesy: Sutton and Barto, 2018

Upper confidence bound (UCB) action selection



- In ε-greedy, the random action selection gives same preference to all actions
  - However, the goal of the non-greedy action is to *explore*
  - Learn the estimates of all actions better
  - At any step t, the uncertainty about all actions may not be the same
  - Also, not all actions have the same potential to be maximal

Upper confidence bound (UCB) action selection

- Choose action based on the potential of that being maximal

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

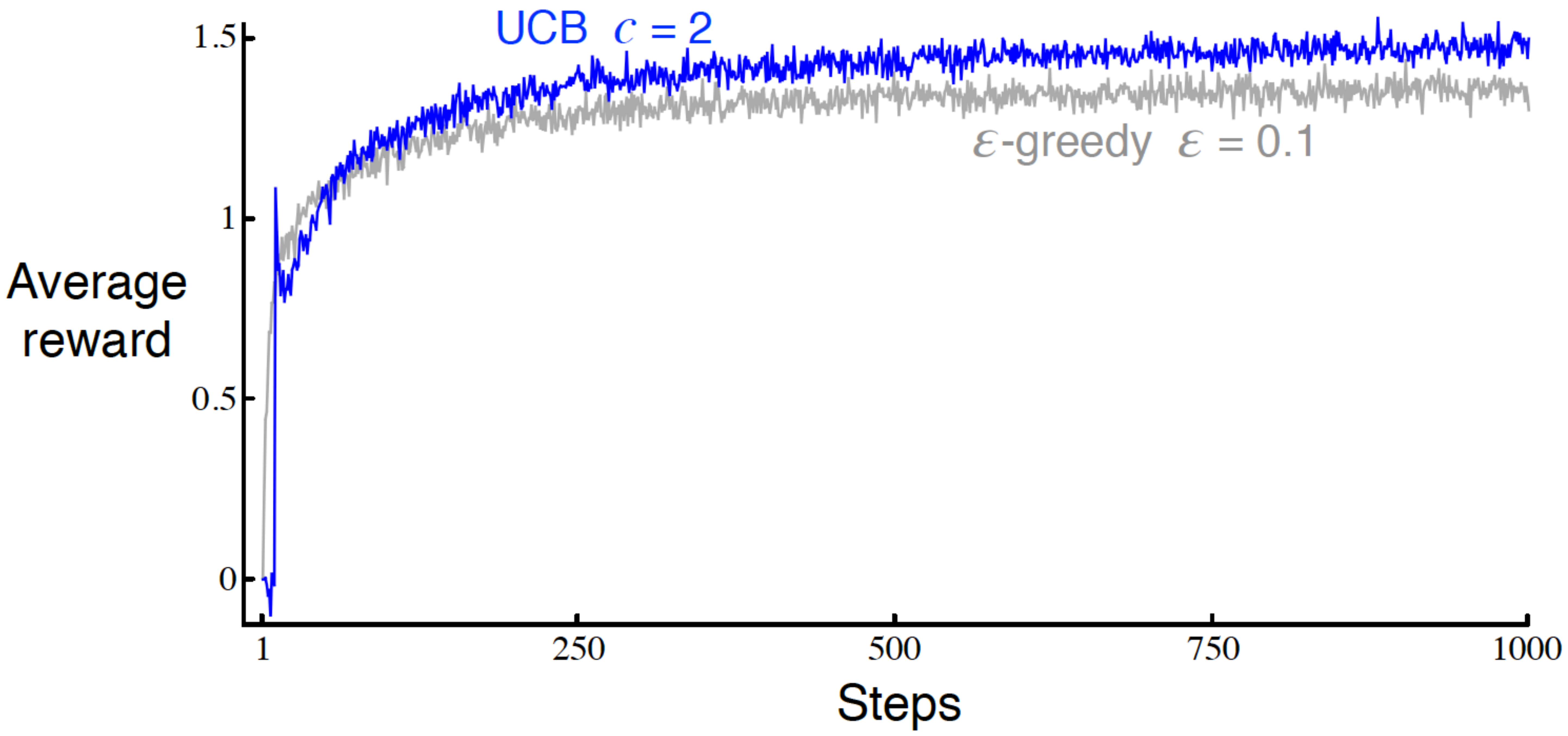
Already estimated value.
More the estimate more the potential.

A measure of uncertainty of $Q_t(a)$.
If $N_t(a)$ is large, action a has been taken many times already.

c : Exploration factor.
$c = 0$ is equivalent to greedy.

Upper confidence bound (UCB) action selection

- UCB performs better than simple ε-greedy in case of the k-armed bandit problem
- However, it is difficult to apply to more general reinforcement learning problems



Picture courtesy: Sutton and Barto, 2018

Gradient ascent method

- Preference scores $H_t(a)$ towards action a
  - Variables: these are what we keep updating
- Choose actions based on a probability
  - Convert the preference scores to probability distribution using softmax
  - Define the probability of choosing action a as $\pi_t(a) = P[A_t = a] = \dfrac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}}$
- Objective function to be maximized: expected reward $\displaystyle\sum_{x=1}^{k} \pi(x)\ q_*(x)$
- But we don't know $q_*(x)$, instead we can draw samples for $x = a$
  - High reward for action a → estimate for $q_*(a)$ increases → should increase $\pi(a)$
  - Objective function increases → parameter $H_t(a)$ increases
  - **Stochastic gradient ascent**

Bandit algorithm as gradient ascent

Our update rule

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

where

$$\mathbb{E}[R_t] = \sum_x \pi_t(x) q_*(x)$$

Derivation of stochastic gradient ascent

- Claim: $\dfrac{\partial \pi_t(x)}{\partial H_t(a)} = \pi_t(x)(\mathbf{1}_{a=x} - \pi_t(a))$, where $\mathbf{1}_{a=x} = 1$ if $a = x$, 0 otherwise

- Proof: $\dfrac{\partial \pi_t(x)}{\partial H_t(a)} = \dfrac{\partial}{\partial H_t(a)} \left[\dfrac{e^{H_t(x)}}{\sum_{y=1}^{k} e^{H_t(y)}} \right] = \dfrac{\frac{\partial e^{H_t(x)}}{\partial H_t(a)} \sum_{y=1}^{k} e^{H_t(y)} - e^{H_t(x)} \frac{\partial \sum_{y=1}^{k} e^{H_t(y)}}{\partial H_t(a)}}{\left(\sum_{y=1}^{k} e^{H_t(y)} \right)^2}$

$$= \frac{\mathbf{1}_{a=x} e^{H_t(x)} \sum_{y=1}^{k} e^{H_t(y)} - e^{H_t(x)} e^{H_t(a)}}{\left(\sum_{y=1}^{k} e^{H_t(y)} \right)^2} = \frac{\mathbf{1}_{a=x} e^{H_t(x)}}{\sum_{y=1}^{k} e^{H_t(y)}} - \frac{e^{H_t(x)} e^{H_t(a)}}{\left(\sum_{y=1}^{k} e^{H_t(y)} \right)^2}$$

$$= \mathbf{1}_{a=x}\, \pi_t(x) - \pi_t(x)\pi_t(a) \;= \pi_t(x)(\mathbf{1}_{a=x} - \pi_t(a))$$

Derivation of stochastic gradient ascent

- Now we investigate the gradient $\dfrac{\partial\, \mathbb{E}[R_t]}{\partial\, H_t(a)}$ in our gradient ascent method
- We have $\dfrac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \dfrac{\partial}{\partial H_t(a)}\left[\sum_x \pi_t(x) q_*(x)\right] = \sum_x q_*(x) \dfrac{\partial \pi_t(x)}{\partial H_t(a)}$

$= \sum_x q_*(x) \dfrac{\partial \pi_t(x)}{\partial H_t(a)} - \sum_x \dfrac{\partial \pi_t(x)}{\partial H_t(a)}$, because the sum $\sum_x \pi_t(x) = 1$, so its gradient is zero

- So, we can write the expression as $\sum_x q_*(x) \dfrac{\partial \pi_t(x)}{\partial H_t(a)} - B_t \sum_x \dfrac{\partial \pi_t(x)}{\partial H_t(a)}$ for any B_t independent of x
- Call B_t the baseline reward
- Hence, the gradient can be written as $\sum_x (q_*(x) - B_t) \dfrac{\partial \pi_t(x)}{\partial H_t(a)}$

Derivation of stochastic gradient ascent

- Next we multiply each term of the sum by $\frac{\pi_t(x)}{\pi_t(x)}$ and obtain the gradient as

$$\sum_x \pi_t(x)(q_*(x) - B_t)\frac{\partial \pi_t(x)}{\partial H_t(a)} / \pi_t(x)$$

- This is in the form of an expectation, sum over all values of x, each term multiplied with $\pi_t(x)$
- So, it is equivalent to $\mathbb{E}\left[\left(q_*(A_t) - B_t\right)\frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right]$
- We don't know $q_*(A_t)$, but we have, $\mathbb{E}[R_t \mid A_t] = q_*(A_t)$
- Hence we use R_t as a replacement for $q_*(A_t)$ at timestep t
- As baseline B_t, one possible option is to take $\bar{R}_t$, average of all rewards so far

Derivation of stochastic gradient ascent

- Then we can write $\mathbb{E}\left[\left(q_*(A_t) - B_t\right)\frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right] = \mathbb{E}\left[(R_t - \bar{R}_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right]$
- Since we know $\frac{\partial\, \pi_t(x)}{\partial\, H_t(a)} = \pi_t(x)(\mathbf{1}_{a=x} - \pi_t(a))$, we can write the gradient as

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

- Our stochastic gradient step: use *sample* of the expectation in place of the gradient

$$H_{t+1}(a) := H_t(a) + \alpha\frac{\partial\, \mathbb{E}[R_t]}{\partial\, H_t(a)} \simeq H_t(a) + \alpha(R_t - \bar{R}_t)(\mathbf{1}_{a=A_t} - \pi_t(a))$$

Gradient ascent : intuition

- Gradient ascent: $H_{t+1}(a) := H_t(a) + \alpha(R_t - \bar{R}_t)(\mathbf{1}_{a=A_t} - \pi_t(a))$
- At each step t, after taking action $a = A_t$ and getting reward R_t, update:

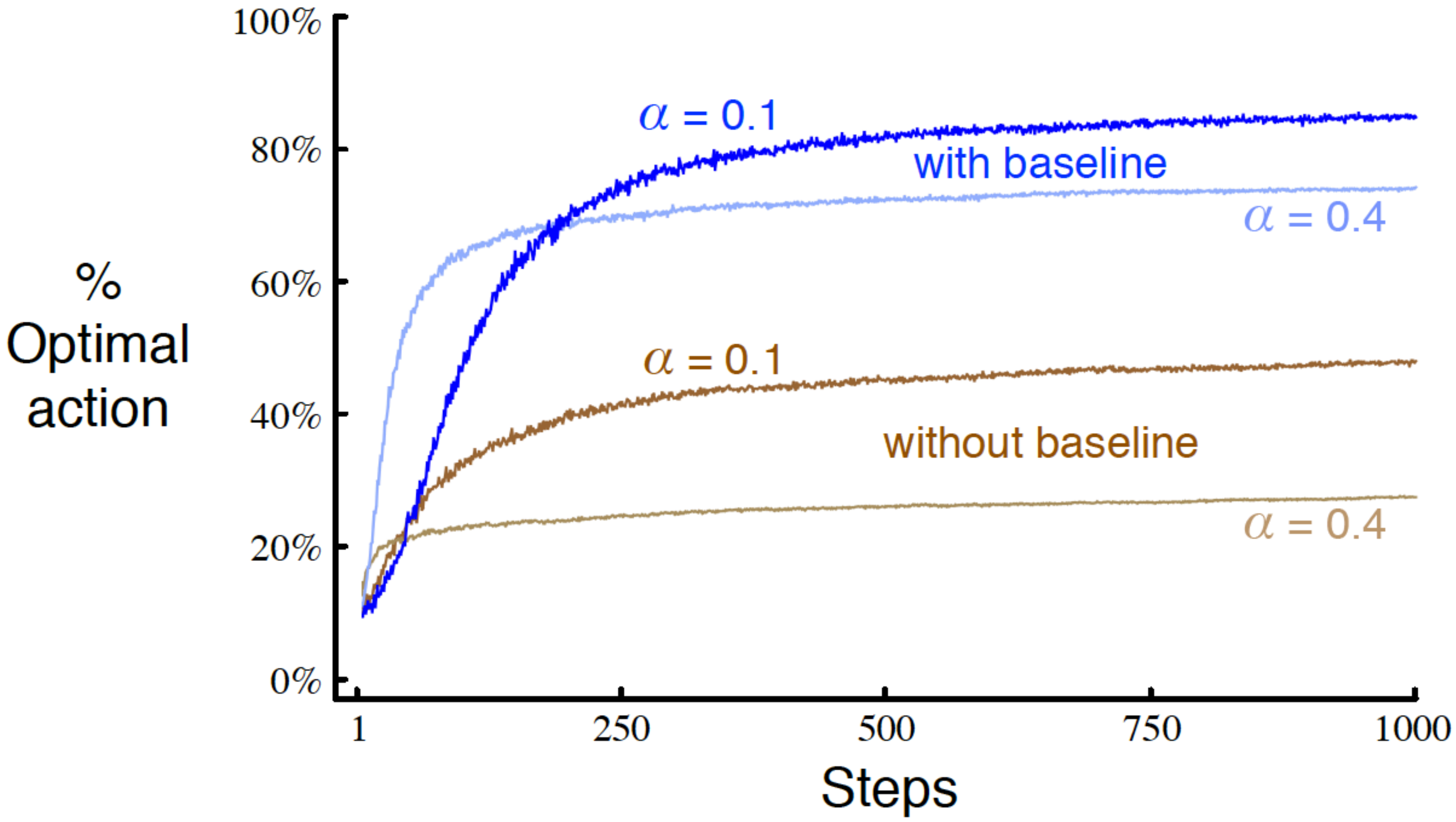
  $H_{t+1}(A_t) := H_t(A_t) + \alpha(R_t - \bar{R}_t)(1 - \pi_t(A_t))$, and

  $H_{t+1}(a) = H_t(a) - \alpha(R_t - \bar{R}_t)\pi_t(a)$ for all $a \neq A_t$
- If reward $R_t >$ baseline $\bar{R}_t$, $H_t(A_t)$ is increased, otherwise decreased
- For all other actions, the opposite happens
- Choice of baseline B_t
  - Irrespective of the choice, as long as it is independent of the action (x), the algorithm would be a stochastic gradient ascent
  - However, the choice affects the variance and affects how the algorithm would converge

Picture courtesy: Sutton and Barto, 2018

Gradient ascent : performance

- Performance on the 10-armed testbed (refer to the book)
- With baseline ($B_t = \bar{R}_t$) and without baseline ($B_t = 0$)



Picture courtesy: Sutton and Barto, 2018

References

- Sutton, R.S. and Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press. http://incompleteideas.net/book/the-book.html (primary reference)
- Ashwin Rao. Multi-armed bandits: exploration vs exploitation. http://web.stanford.edu/class/cme241/lecture_slides/MultiArmedBandits.pdf