In the Name of God

Iran University of Science and Technology

دانشگاه علم و صنعت ایران

Reinforcement Learning in Control

Dr. Saeed Shamaghdari

Electrical Engineering Department
Control Group

Fall 2025 | 4041

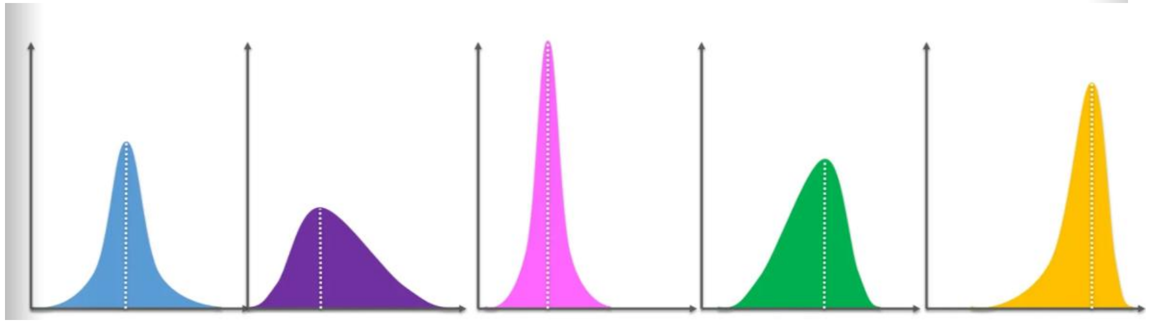
Multi-Armed Bandit

Multi-Armed Bandits

Multi-Armed Bandits

Action: $a_t \in A$
Reward: $r_t \sim R \quad R_a(r) = P(r|a)$
Goal: maximize $\Sigma\, r_t$
Value: $Q(a) \approx E[r|a]$

Exact Value: $q_*(a) = E[r|a]$
Optimal Value: $\max_{a \in A} Q(a)$

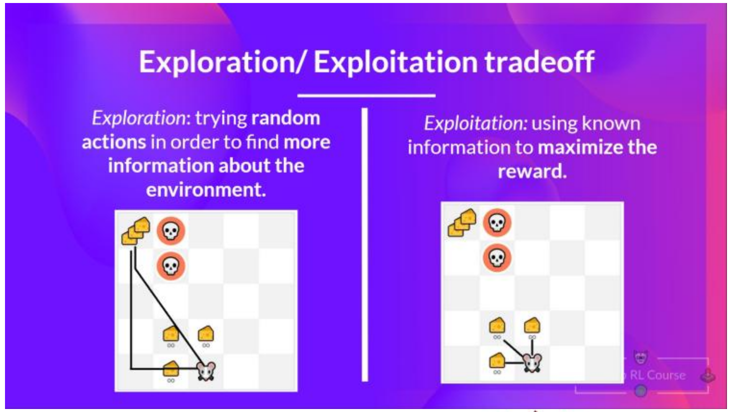


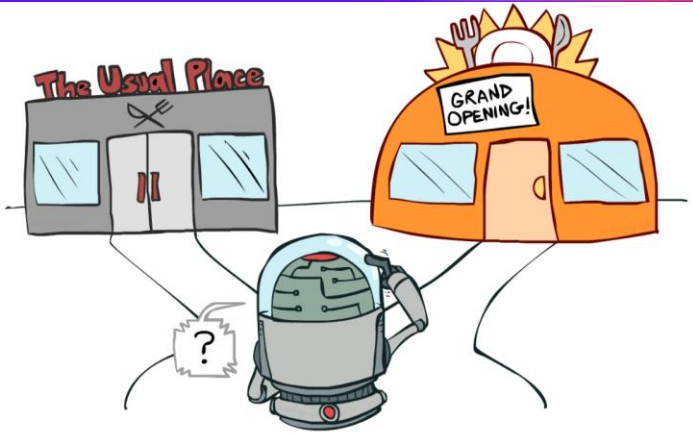
Greedy and ϵ-Greedy Actions

Greedy Action: Exploitation only
Exploration: Selecting a Non-Greedy Action

Exploitation or Exploration??

Action Selection:
Greedy Action: $$A_t = \arg\max Q_t(a)$$
ϵ-Greedy Action





Reminder from Lecture 1:
Exploitation vs Exploration

An Approach to Value Function Estimation

Sample Average:

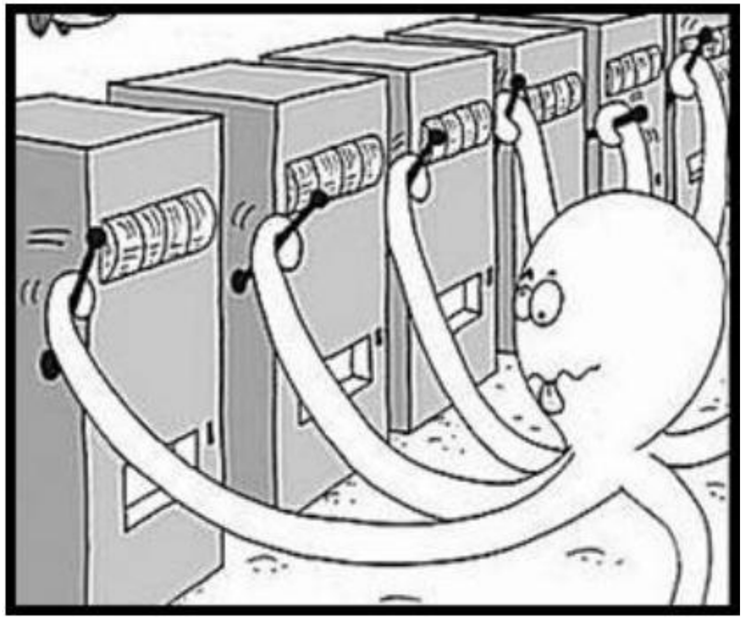
$$Q_t(a) \doteq \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i=a}}$$

Convergence: (by the law of large numbers…)

$$Q_t(a) \rightarrow q_*(a) \qquad \text{for all } a \in A$$

Q: Under convergence conditions, what is the probability of choosing the optimal action in the ϵ-greedy policy?

The 10-Armed Testbed



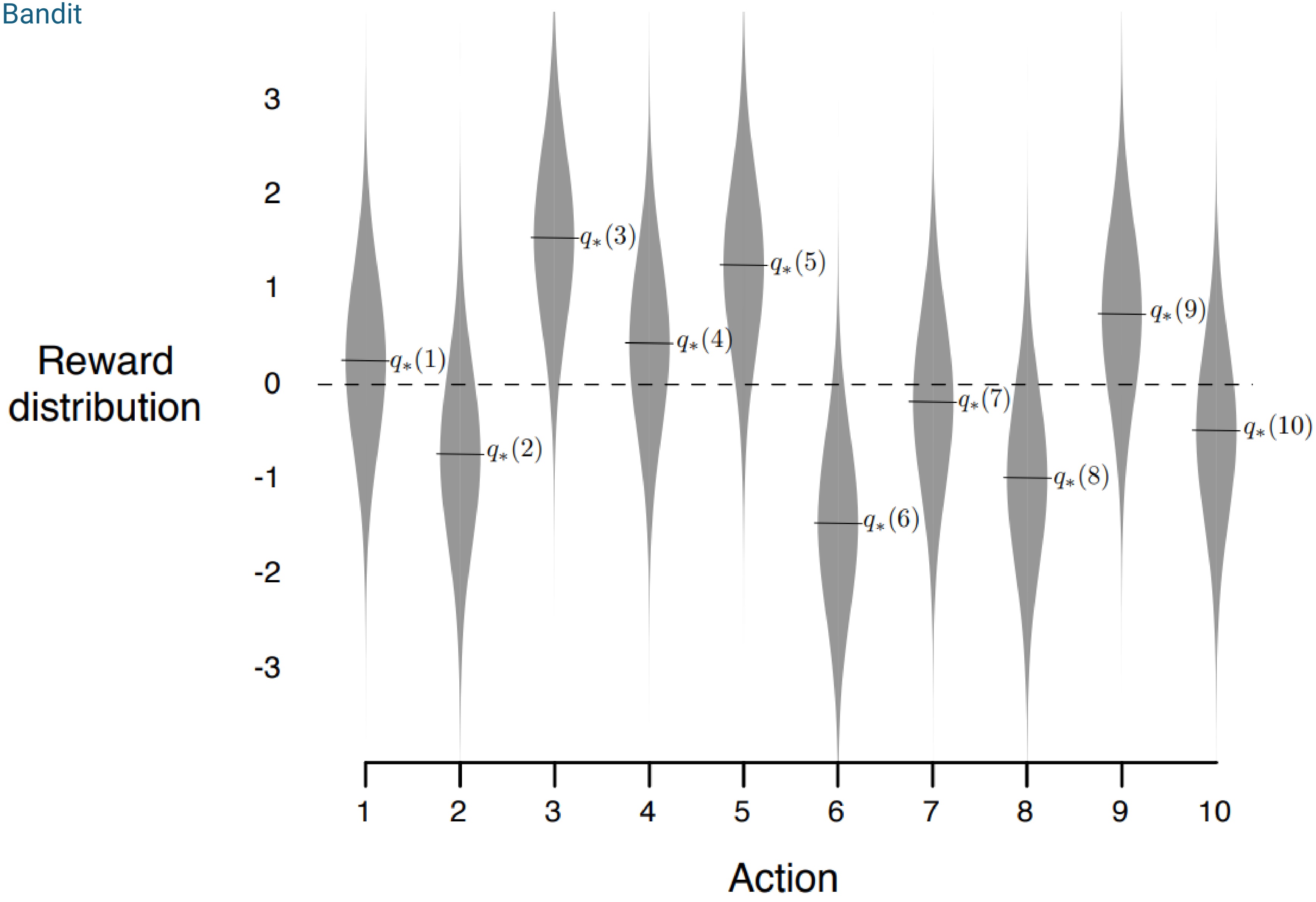
One Run? 1000 sample
Average Behavior?
Average in 2000 problem solution

Generating values of $q_*(a)$ for each action a:

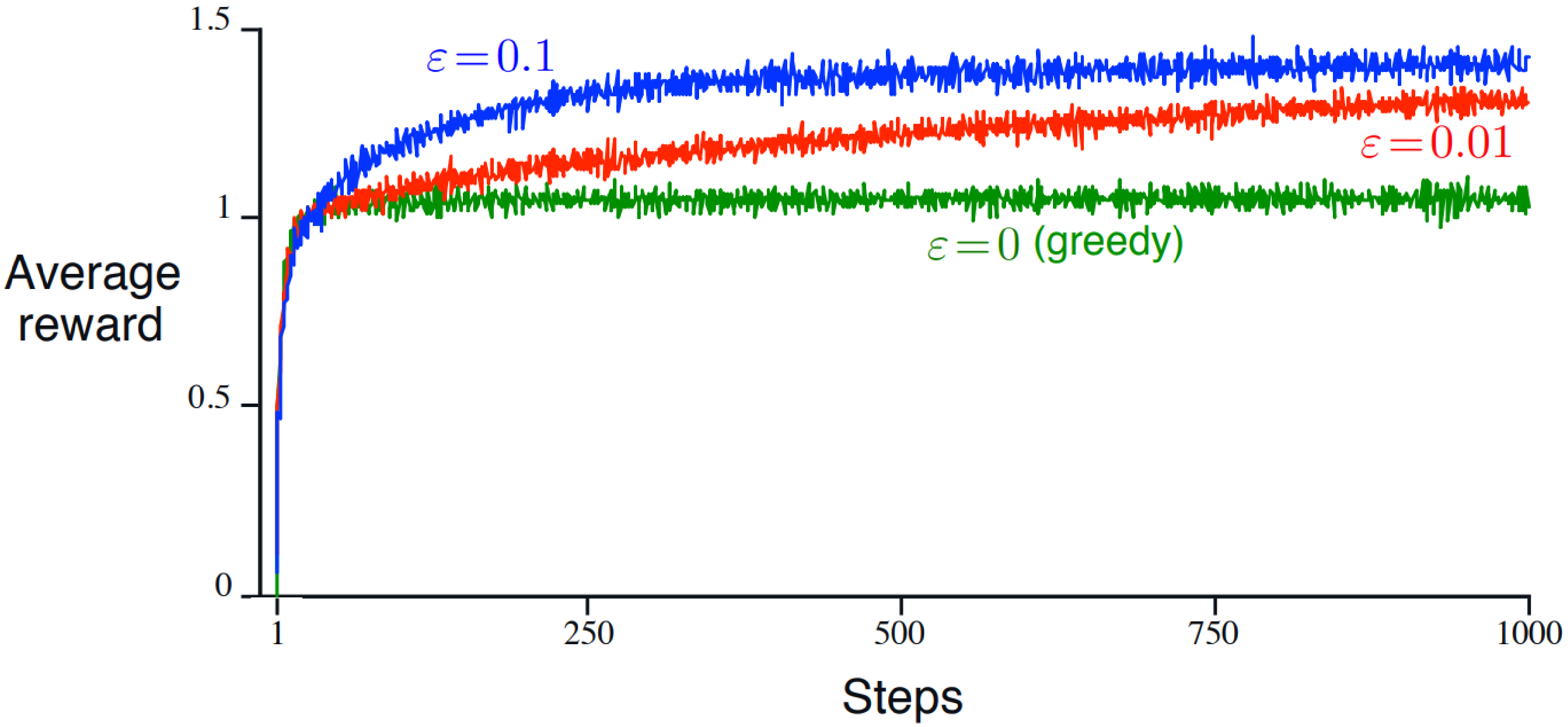
Normal distribution with mean 0 and variance 1

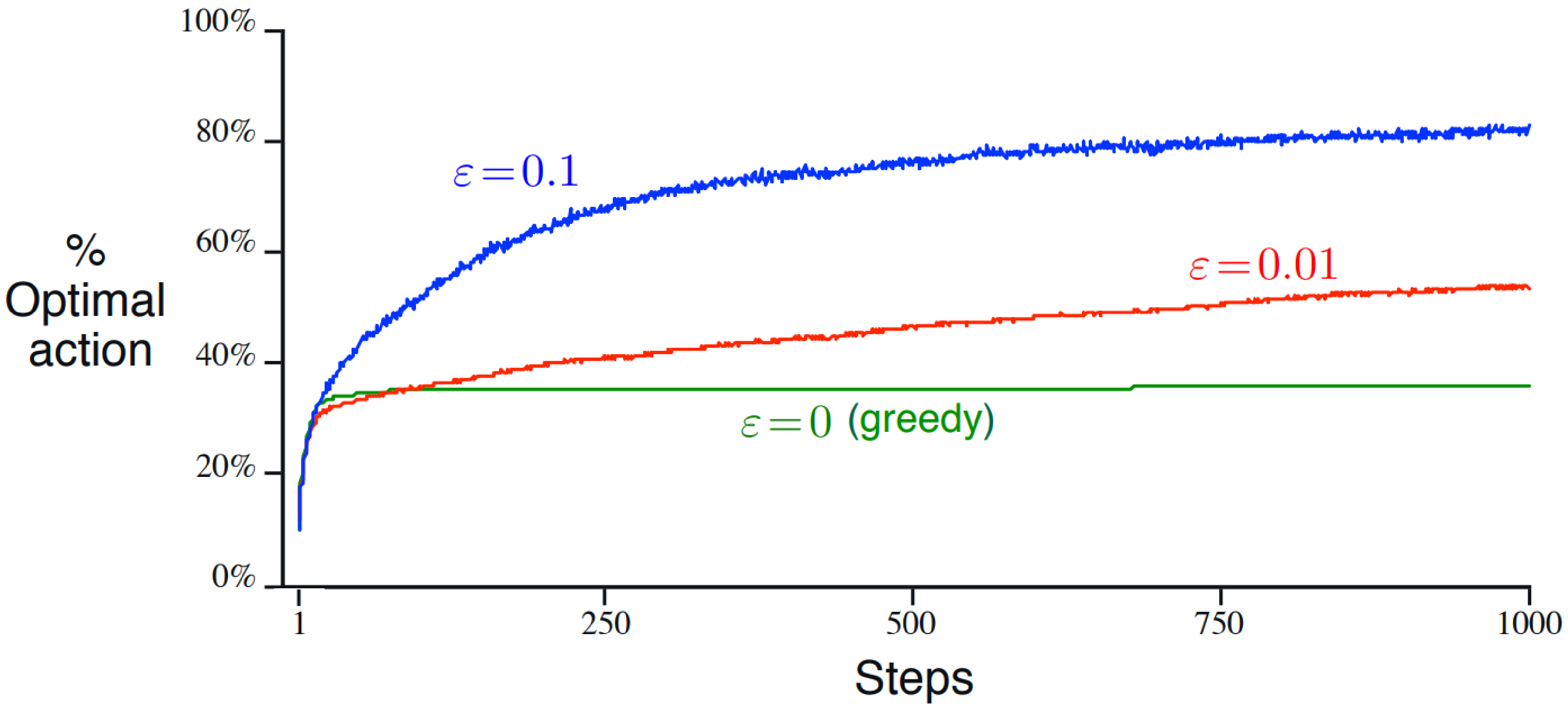
Reward values for each action a:

Normal distribution with mean $q_*(a)$ and variance 1



Reward distribution
3
2
1
0
-1
-2
-3
$q_*(1)$
$q_*(2)$
$q_*(3)$
$q_*(4)$
$q_*(5)$
$q_*(6)$
$q_*(7)$
$q_*(8)$
$q_*(9)$
$q_*(10)$
1
2
3
4
5
6
7
8
9
10
Action


1.5
1
0.5
0
Average reward
$\varepsilon = 0.1$
$\varepsilon = 0.01$
$\varepsilon = 0$ (greedy)
1
250
500
750
1000
Steps


100%
80%
60%
40%
20%
0%
% Optimal action
$\varepsilon = 0.1$
$\varepsilon = 0.01$
$\varepsilon = 0$ (greedy)
1
250
500
750
1000
Steps

Incremental Implementation

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

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

Incremental Implementation

$$NewEstimate \leftarrow OldEstimate + \underbrace{StepSize}_{\alpha} \underbrace{\left[Target - OldEstimate\right]}_{\text{Error}}$$

(Proportional to exploration)

Addressing Non-Stationarity in Reinforcement Learning

Greater weight given to recent rewards

Replace sample average with a weighted average

Assuming a constant value for $\alpha \in (0,1]$:

$$Q_{n+1} \doteq Q_n + \alpha \Big[R_n - Q_n \Big]$$

Addressing Non-Stationarity in Reinforcement Learning

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha)Q_n \\
&= \alpha R_n + (1-\alpha)\left[\alpha R_{n-1} + (1-\alpha)Q_{n-1}\right] \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1} \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 \alpha R_{n-2} + \\
&\qquad \cdots + (1-\alpha)^{n-1}\alpha R_1 + (1-\alpha)^n Q_1 \\
&= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i .
\end{aligned}
$$

Addressing Non-Stationarity in Reinforcement Learning

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha)Q_n \\
&= \alpha R_n + (1-\alpha)\left[\alpha R_{n-1} + (1-\alpha)Q_{n-1}\right] \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1} \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 \alpha R_{n-2} + \\
&\qquad \cdots + (1-\alpha)^{n-1}\alpha R_1 + (1-\alpha)^n Q_1 \\
&= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i .
\end{aligned}
$$

Q: How does increasing α affect the variance of the estimate?

Weighted Average

Unbiased Estimation: $if\ n \to \infty \quad then\ Q = E[R(a)] = q_*(a)$

Addressing Non-Stationarity in Reinforcement Learning

Weighted average with time-varying α: $\alpha_n(a)$

Convergence condition

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

The sequence α_n does not decrease
Reduces the effect of the initial condition and fluctuations
Sufficient exploration

The sample average case: $\alpha_n(a) = \frac{1}{n}$

Note: This condition does **not** hold for every constant step-size

Addressing Non-Stationarity in Reinforcement Learning

The sample average case: $\alpha_n(a) = \dfrac{1}{n}$

Note: This condition does **not** hold for every constant step-size

$$1 + \frac{1}{2} + \overbrace{\frac{1}{3} + \frac{1}{4}} + \ \ldots \ \geq 1 + \frac{1}{2} + \overbrace{\frac{1}{4} + \frac{1}{4}} + \underbrace{\left(\frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8}\right)} + \cdots + (\frac{1}{16} + \cdots$$

$$\geq 1 + 1 + 1 + \cdots = \infty$$

Addressing Non-Stationarity in Reinforcement Learning

Unbiased estimator:

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

$$Q_n = Q_{n-1} + \alpha_{n-1}[R_{n-1} - Q_{n-1}]$$

$$= (1-\alpha_{n-1})Q_{n-1} + \alpha_{n-1}R_{n-1}$$

$$Q_{n+1} = \alpha_n R_n + (1-\alpha_n)\alpha_{n-1}R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1})\,Q_{n-1}$$

$$= \alpha_n R_n + (1-\alpha_n)\alpha_{n-1}R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1}) \cdot \alpha_{n-2}R_{n-2} + \cdots$$

$$q_* = E[R](\alpha_n + (1-\alpha_n)\alpha_{n-1} + (1-\alpha_n)(1-\alpha_{n-1})\,\alpha_{n-2}$$

$$+ \cdots + (1-\alpha_n)(1-\alpha_{n-1})\cdots(1-\alpha_1))$$

$$q_* = E[R](\alpha_n + (1-\alpha_n)\underbrace{[\alpha_{n-1} + (1-\alpha_{n-1})\alpha_{n-2} + \cdots]}_{S})$$

Addressing Non-Stationarity in Reinforcement Learning

Unbiased estimator:

$$q_* = E[R]\big(\alpha_n + (1-\alpha_n)\alpha_{n-1} + (1-\alpha_n)(1-\alpha_{n-1})\,\alpha_{n-2} + \cdots + (1-\alpha_n)(1-\alpha_{n-1})\cdots(1-\alpha_1)\big)$$

$$q_* = E[R]\big(\alpha_n + (1-\alpha_n)\underbrace{[\alpha_{n-1} + (1-\alpha_{n-1})\alpha_{n-2} + \cdots]}_{S}\big)$$

$$S = \alpha_n + (1-\alpha_n)S$$

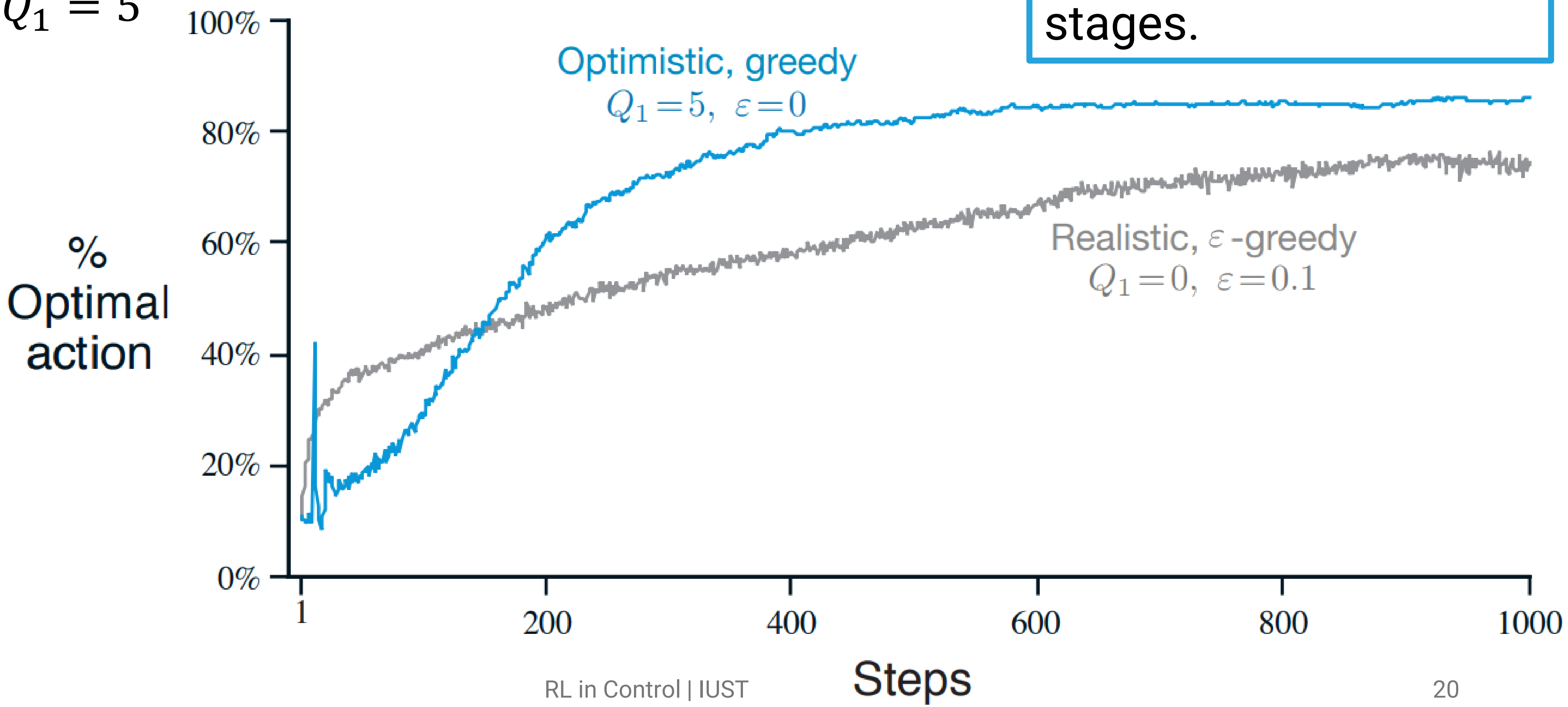
$$(1+\alpha_n)S = S + \alpha_n \rightarrow S = 1$$

Optimistic Initial Values

Choosing optimistic initial values:

Example: $Q_1 = 5$

The need for sufficient exploration in the early stages.



Upper Confidence Bound

Uncertainty in Value Estimation: The Need for Exploration

Q: Drawback of the ε-Greedy Strategy?

In the ε-greedy method, exploration is performed randomly among the non-greedy actions.

It would be more appropriate if the probability of each non-greedy action being optimal were also considered during exploration.

Upper Confidence Bound

Incorporating uncertainty in the estimation process!

$$A_t \doteq \underset{a}{\arg\max} \left[Q_t(a) + \overbrace{c\sqrt{\frac{\ln t}{N_t(a)}}}^{\text{Uncertainty}} \right]$$

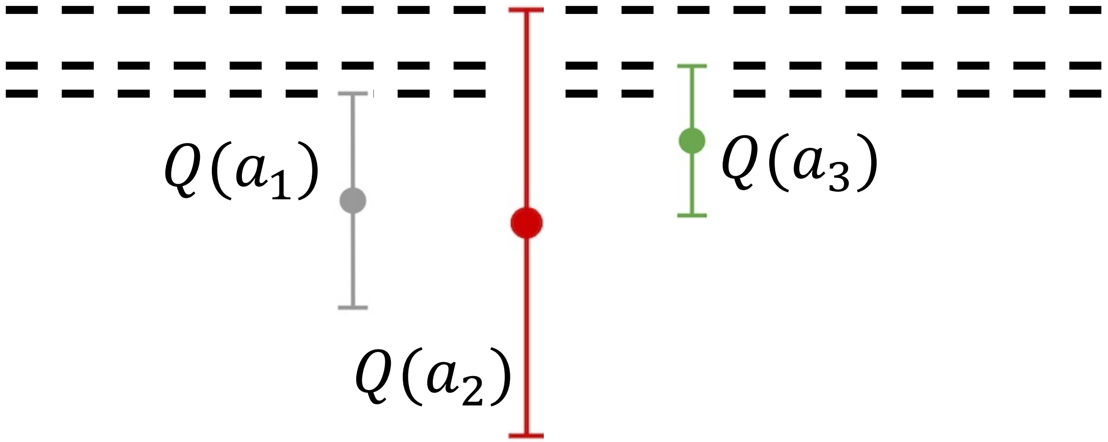
Controlling the level of exploration

Number of times action **a** has been selected before time **t** (**Q**: if zero?)

Each time **a** is selected → Uncertainty decreases
Each time **a** is not selected → Uncertainty increases

Upper Confidence Bound

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



Point estimate >> Interval estimate

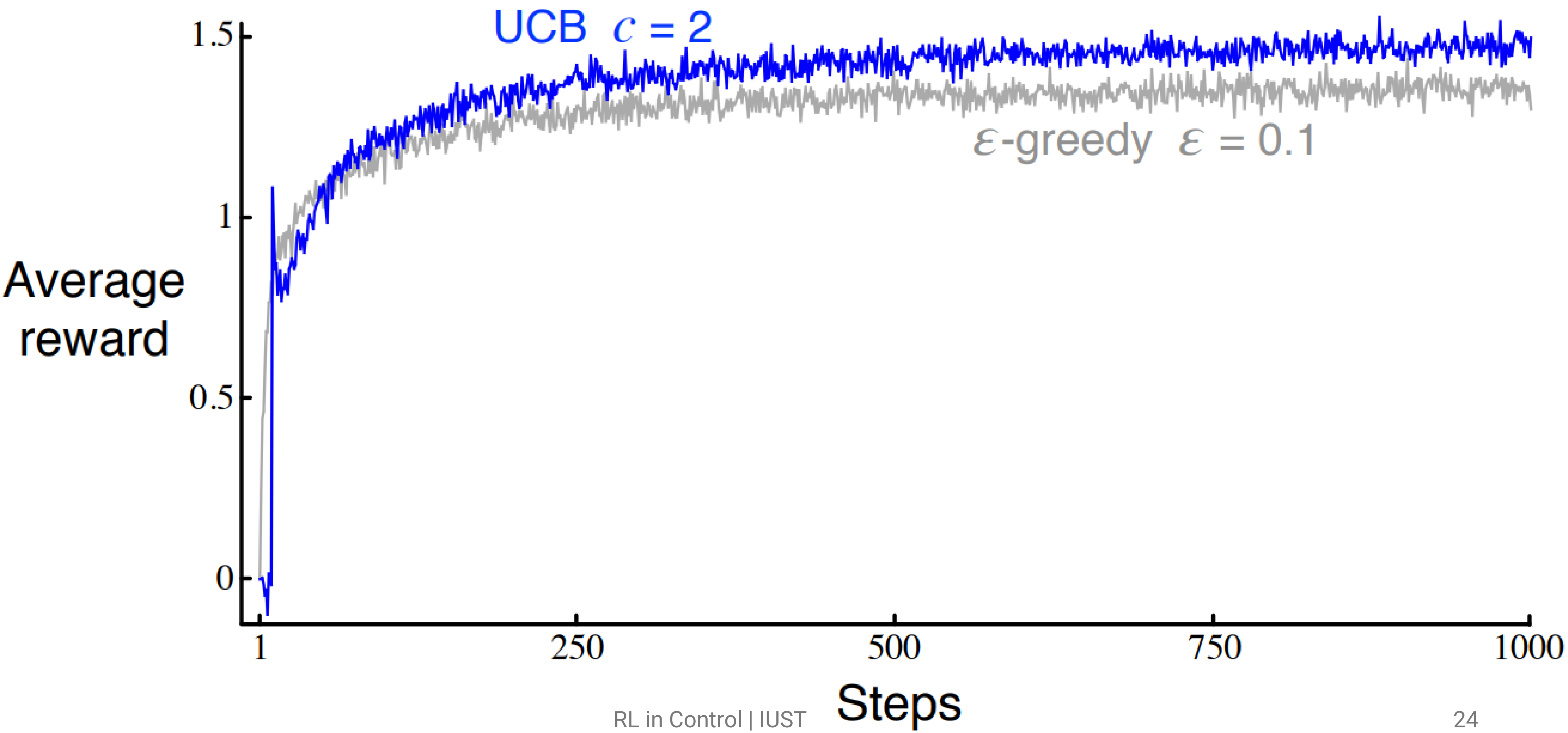
Ensuring all actions are eventually explored

Less frequent selection of actions with low estimated values

More realistic exploration

Automatic and continuous balancing of exploration and exploitation

Upper Confidence Bound



Gradient Bandit Algorithm

We consider learning a numerical preference for each action a (instead of action-value estimation), which we denote $H_t(a)$.

The action probabilities, which are determined according to a soft-max distribution:

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

Gradient Bandit Algorithm

$H_1(a) = 0 \quad for\ all\ a \in \mathcal{A}$: t=1

The action preferences are updated by: (in t>1)

$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha\big(R_t - \bar{R}_t\big)\big(1 - \pi_t(A_t)\big), \qquad \text{and}$$

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

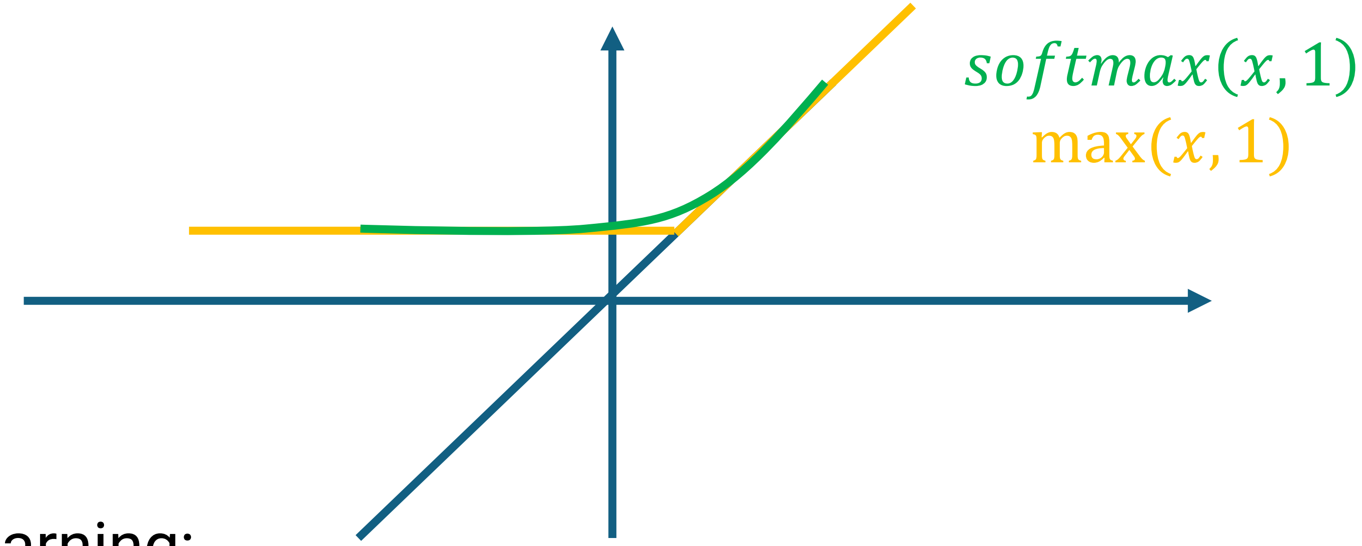
The non-selected actions update in the opposite direction.

$\bar{R}_t \in \mathbb{R}$ is the average of all the rewards up through and including time t

Baseline
- $R_t > \bar{R}_t$ Increasing the probability of selecting **A_t** in the future
- $R_t < \bar{R}_t$ Decreasing the probability of selecting **A_t** in the future

Soft-max Distribution

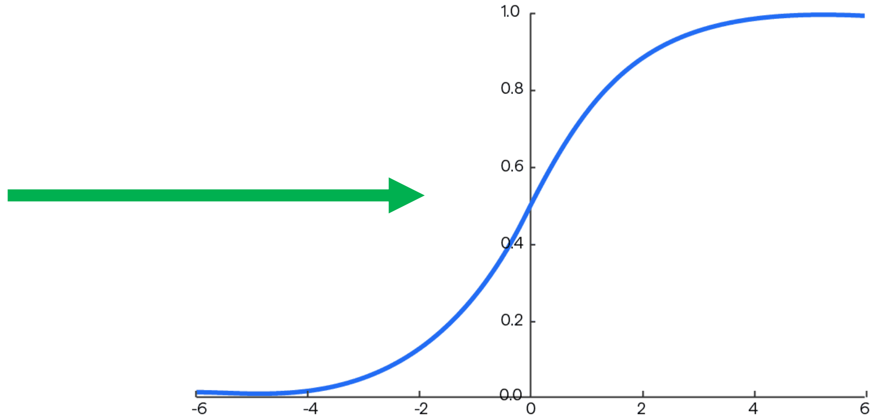
$$softmax(x_1, \dots, x_n) = log \sum_{i=1}^{n} e^{x_i}$$



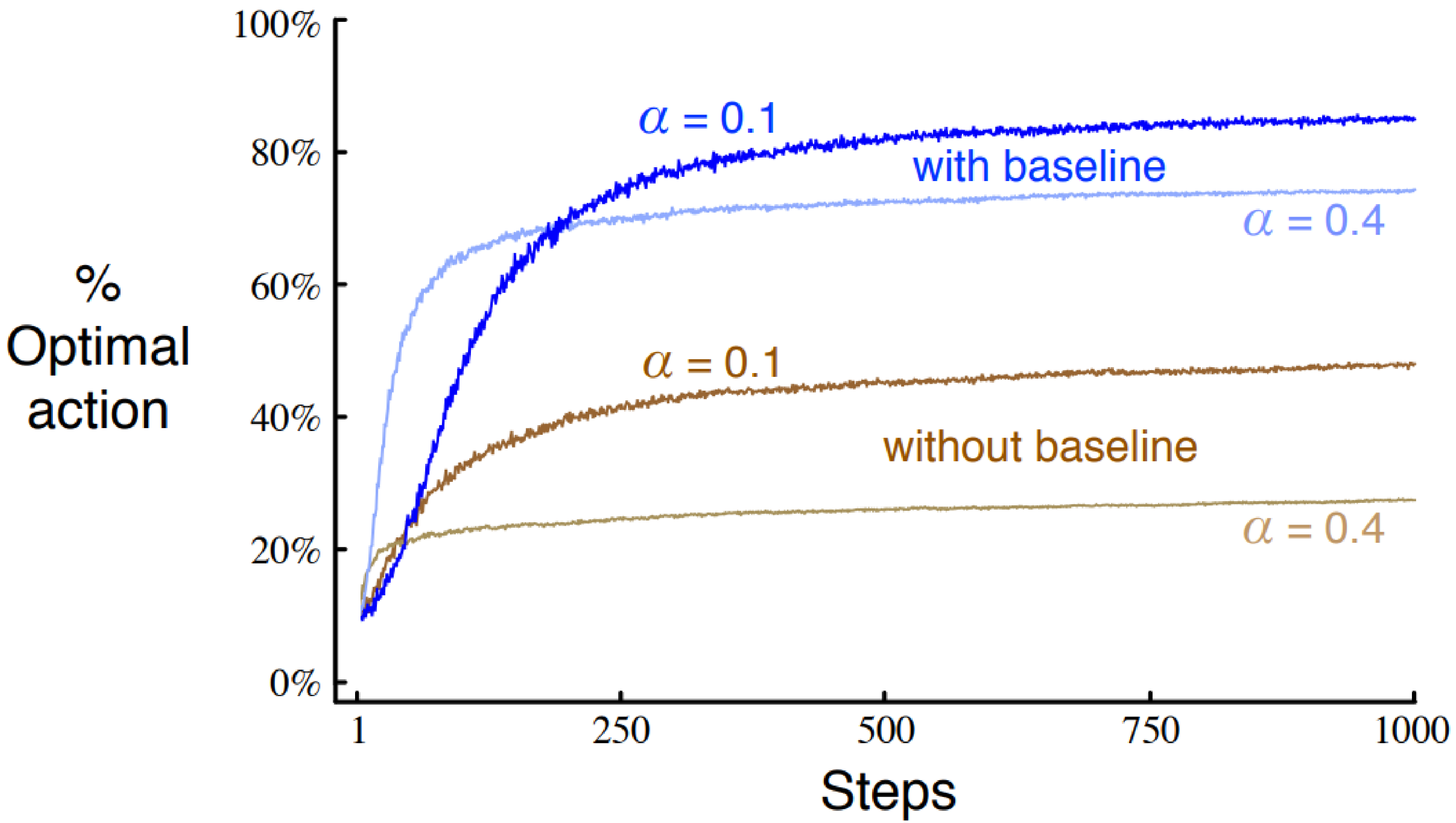
In **Neural Networks and Machine Learning**:

Softmax converts the final layer's outputs into probabilities for classification.

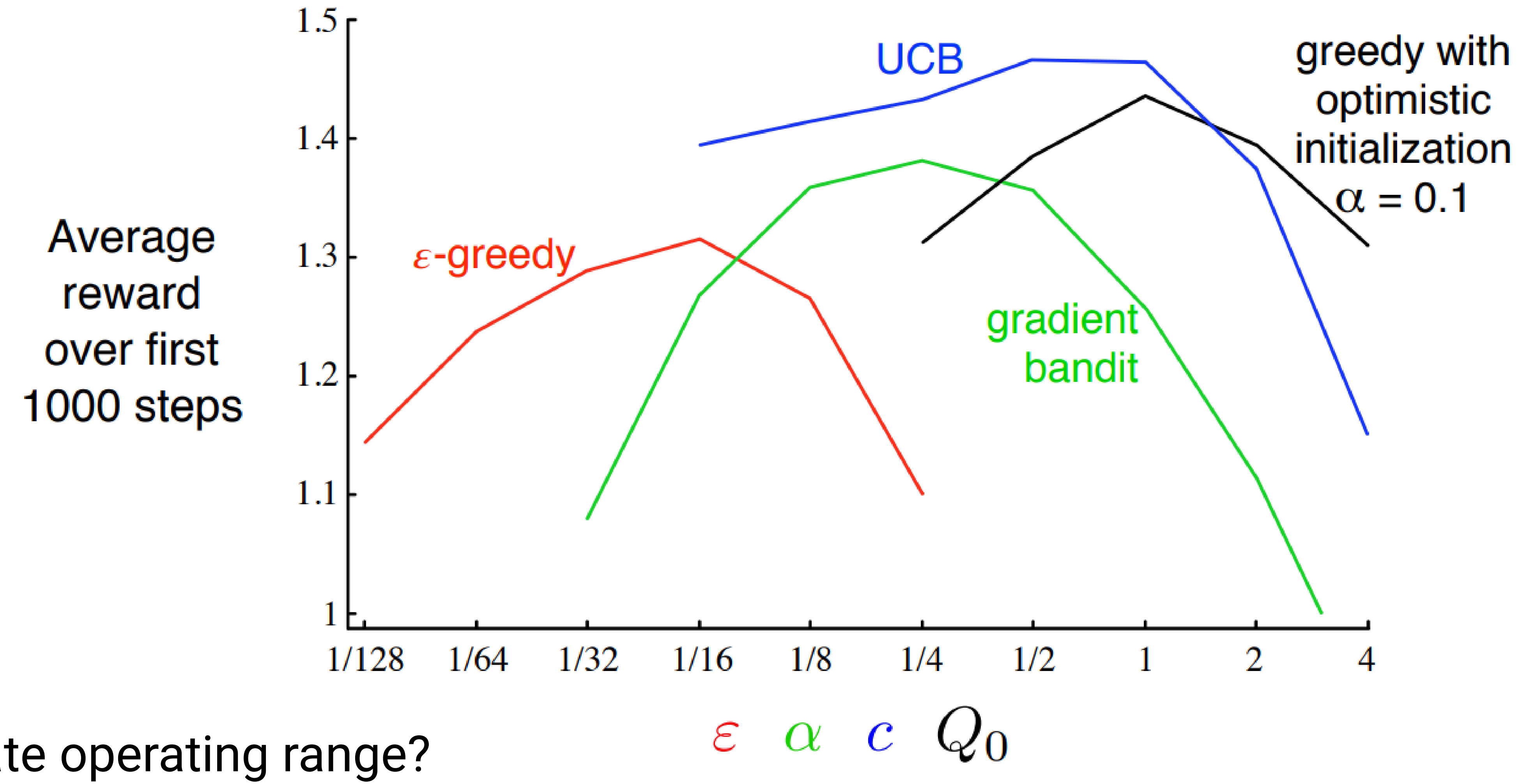
$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$



Gradient Bandit Algorithm



Comparison



➡️ **Q**: Appropriate operating range?
Consider the sensitivity of changes to the parameter