SAPIENZA
UNIVERSITÀ DI ROMA

Autonomous Networking

Gaia Maselli
Dept. of Computer Science

Today's plan

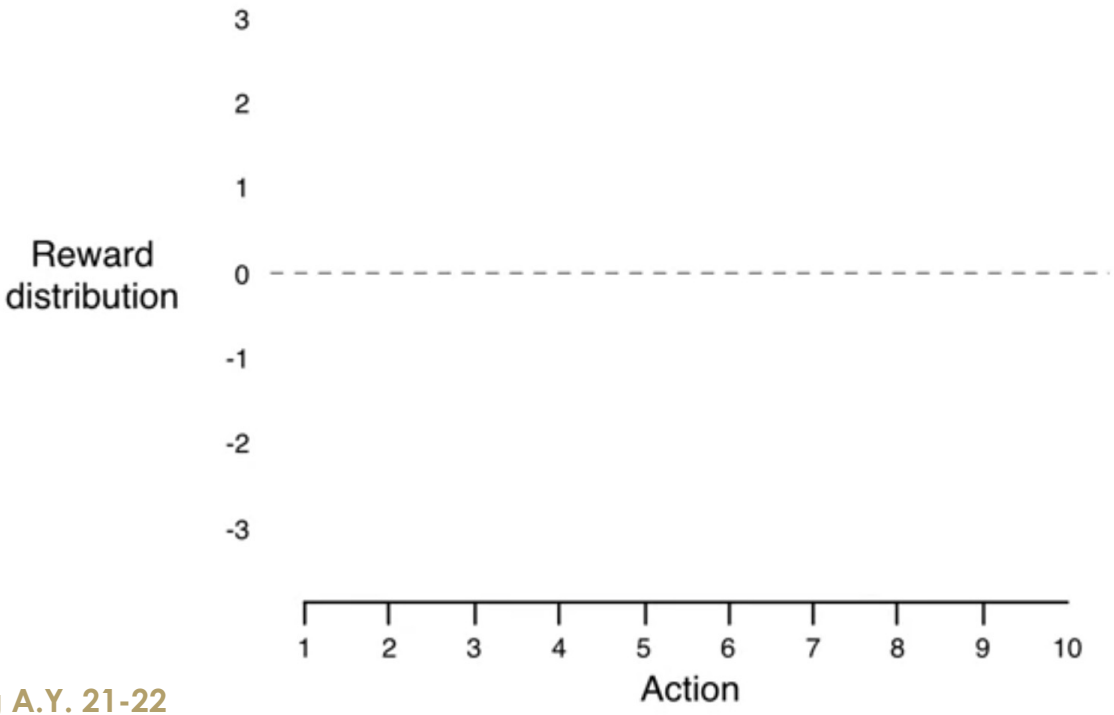
- Performance of methods for action selection
- A new optimistic method

Learning methods

- Strategies for action selection
  - Random
  - Greedy
  - ε-greedy
  - Optimistic initial values
- What is their effectiveness?

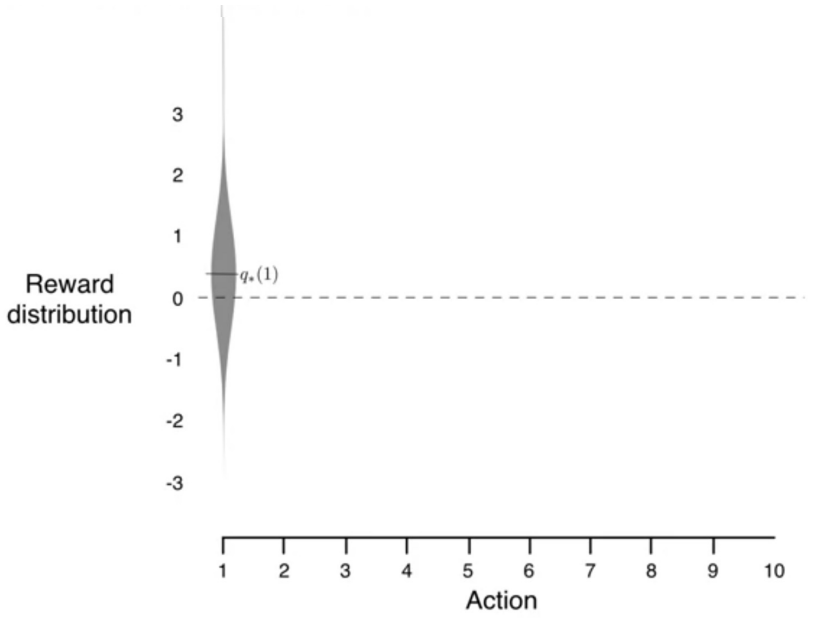
The 10-arms testbed

- To assess the relative **effectiveness** of the different learning methods we compare them numerically
- 10-armed bandit problem (10 actions shown along the X-axis)
- The Y-axis shows the distribution of rewards



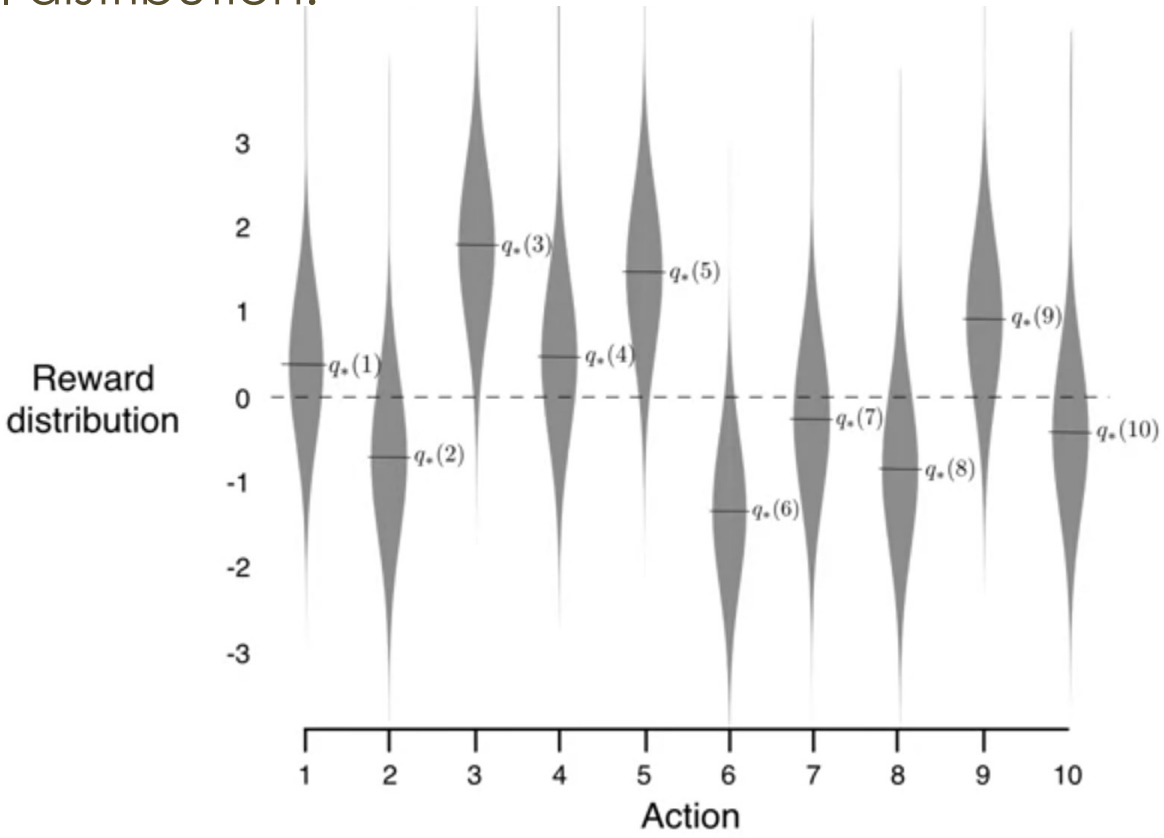
The 10-arms testbed

- Each reward is sampled from a normal distribution with some mean $q_*(a)$ and variance=1
- Each $q_*(a)$ is drawn from a normal distribution with mean=0 and variance=1.



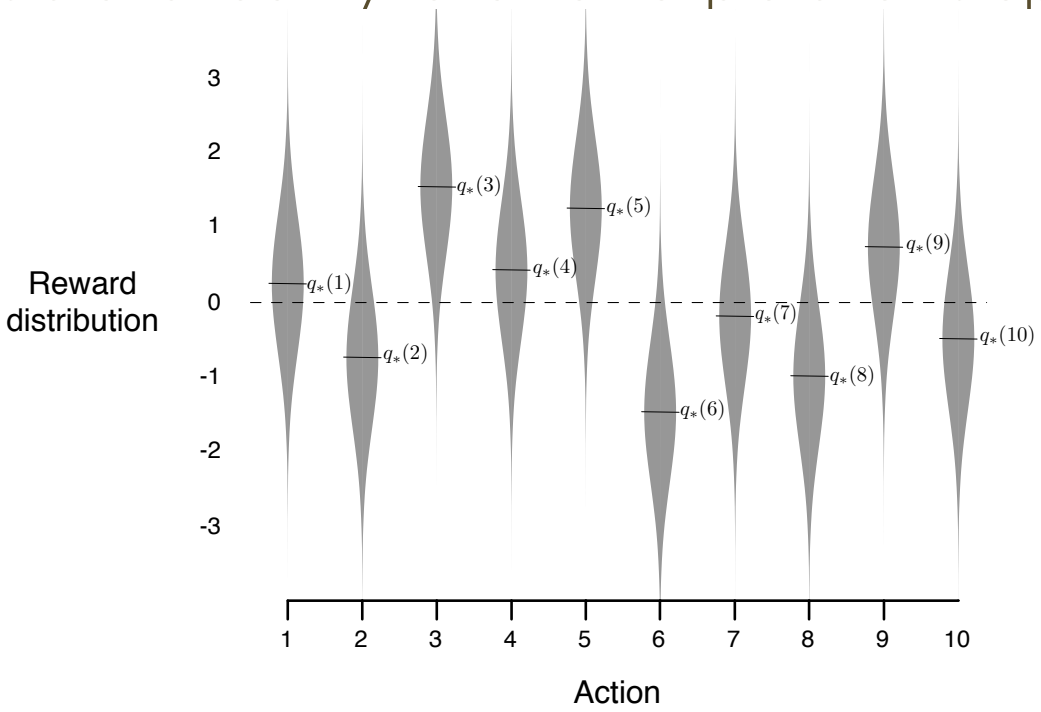
The 10-arms testbed

- Each time we run the 10-arm Testbed q_* will be redrawn from a normal distribution.



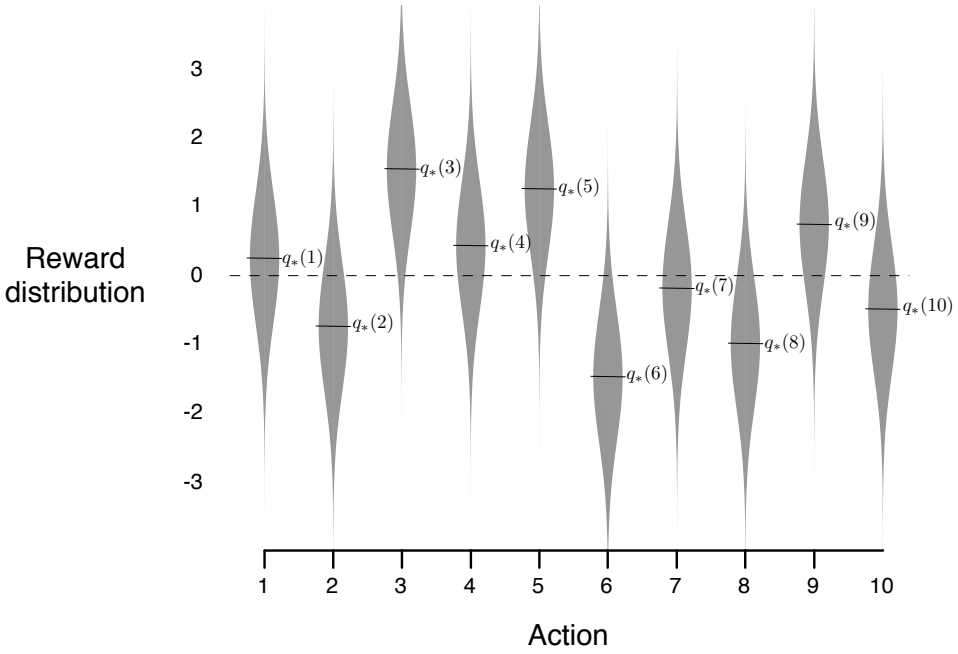
Randomness

- q_* is randomly sampled from a normal distribution
- The rewards are randomly sampled based on q_*
- The actions are randomly taken on exploration steps



The 10-arms testbed

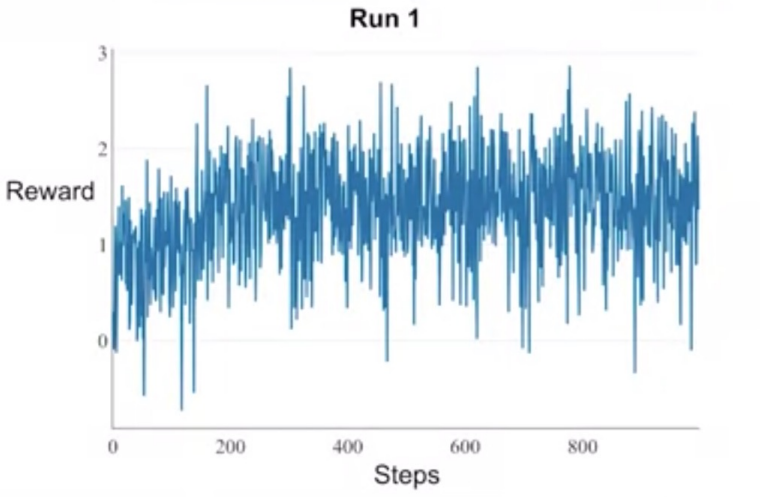
- To fairly compare the different methods we need to perform many independent runs
- For any learning method, we measure its performance over 2000 independet runs

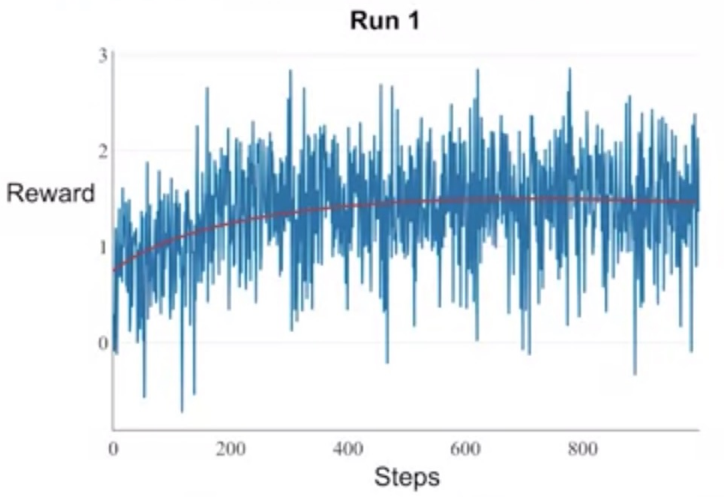


- Each run tests the learning method over 1000 steps
- Random seed
- All the methods form their action-value estimates using the **sample-average** technique

Single run

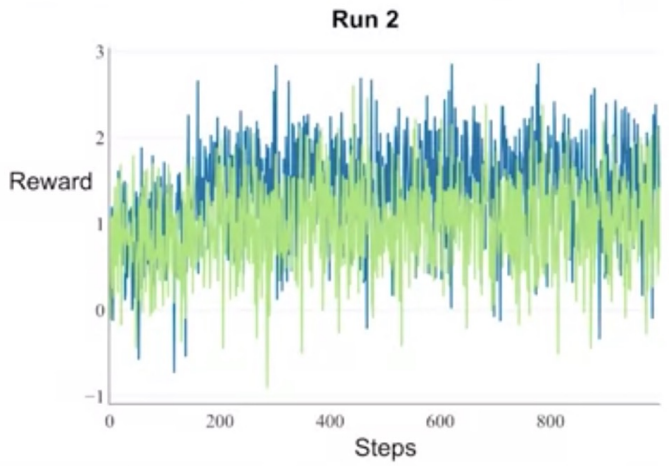
- Single run of an ε-Greedy agent in the 10-arm testbed, with ε=0.1
- The time-step is on the X-axis
- The Y-axis is the reward received on that time-step

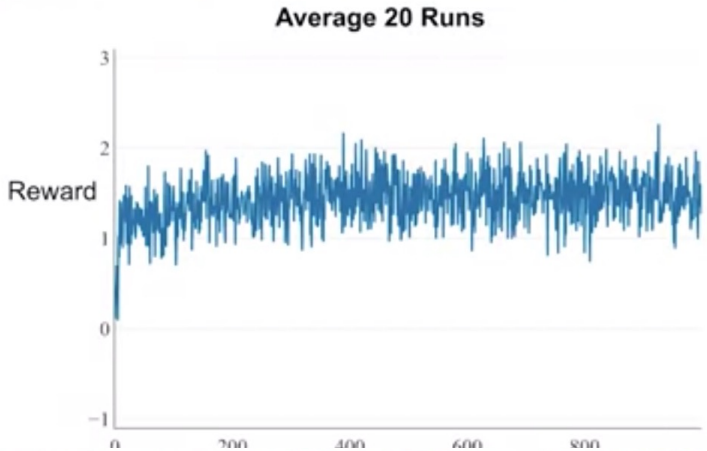


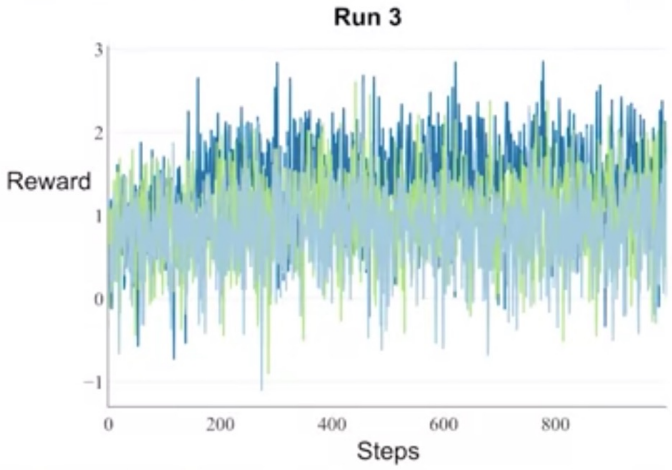


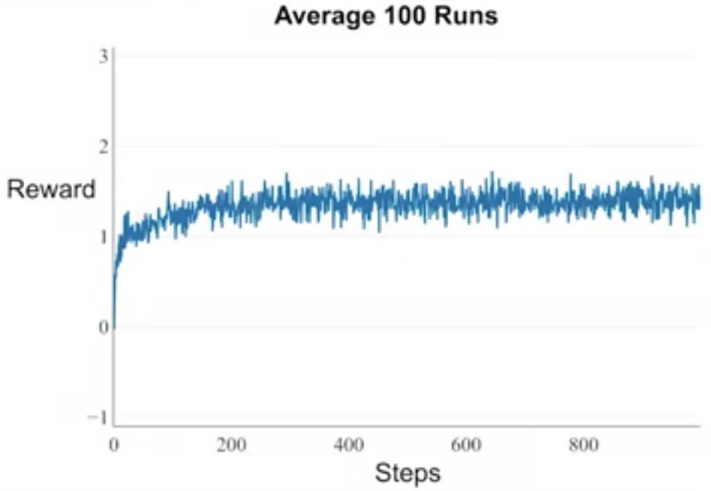
Multiple runs

For every time-step, we can take the average of each of these three rewards





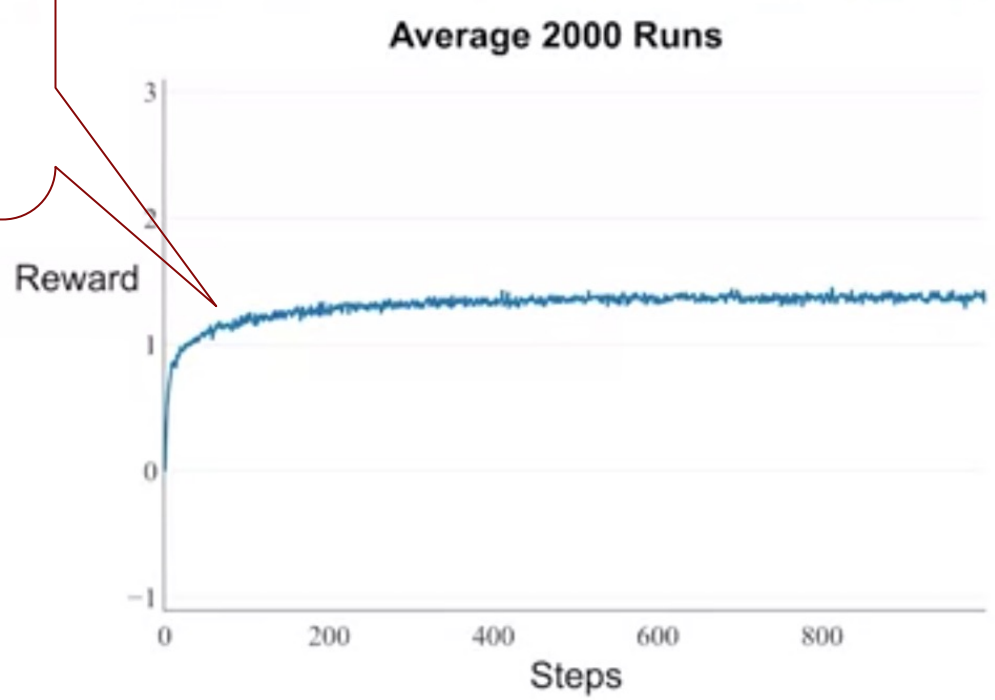




2000 runs

- With 2000 independent runs we obtain a measure of the learning algorithm's average behavior

Noticeable increase
in reward
in the first 200 steps

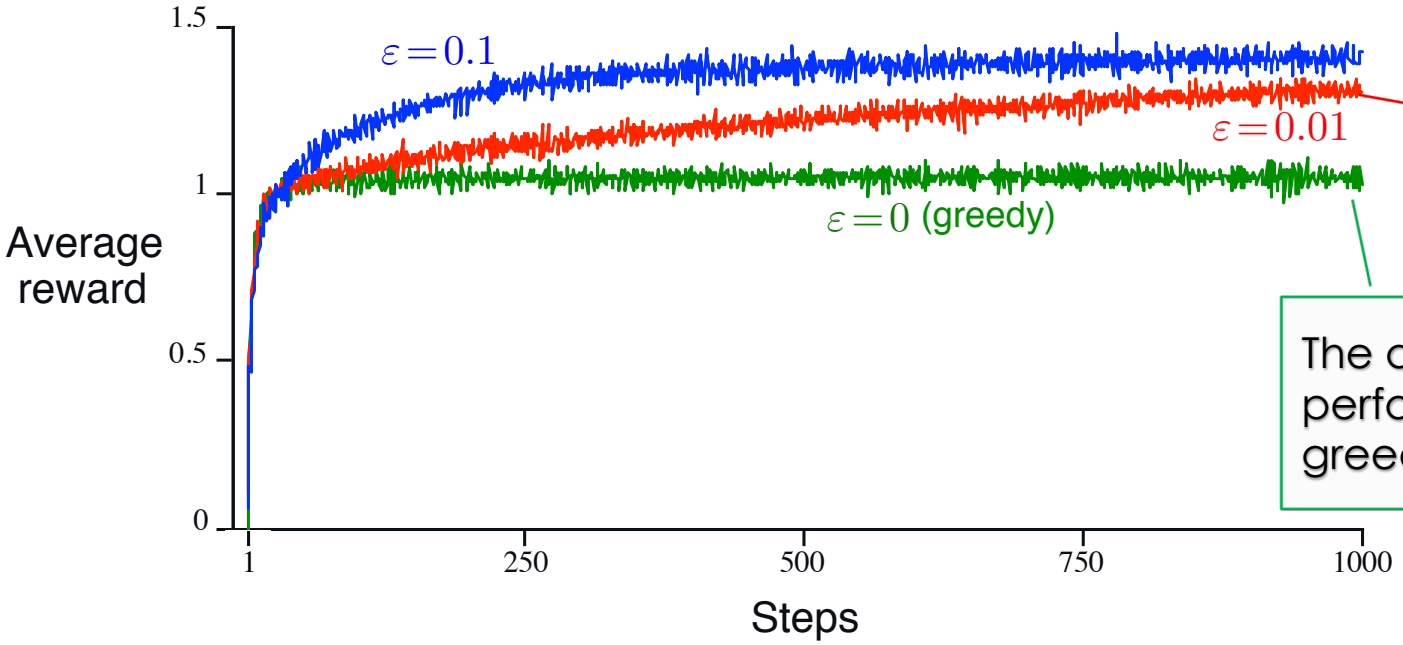


Experiments

- Let us run experiments for different values of ε
  - ε=0 (Greedy)
  - ε=0.01
  - ε=0.1

Performance ε-greedy

- The greedy method achieves a reward-per-step of only about 1, compared with the best possible of about 1.55 on this testbed.
- The greedy method performs significantly worse in the long run because it gets stuck performing suboptimal actions

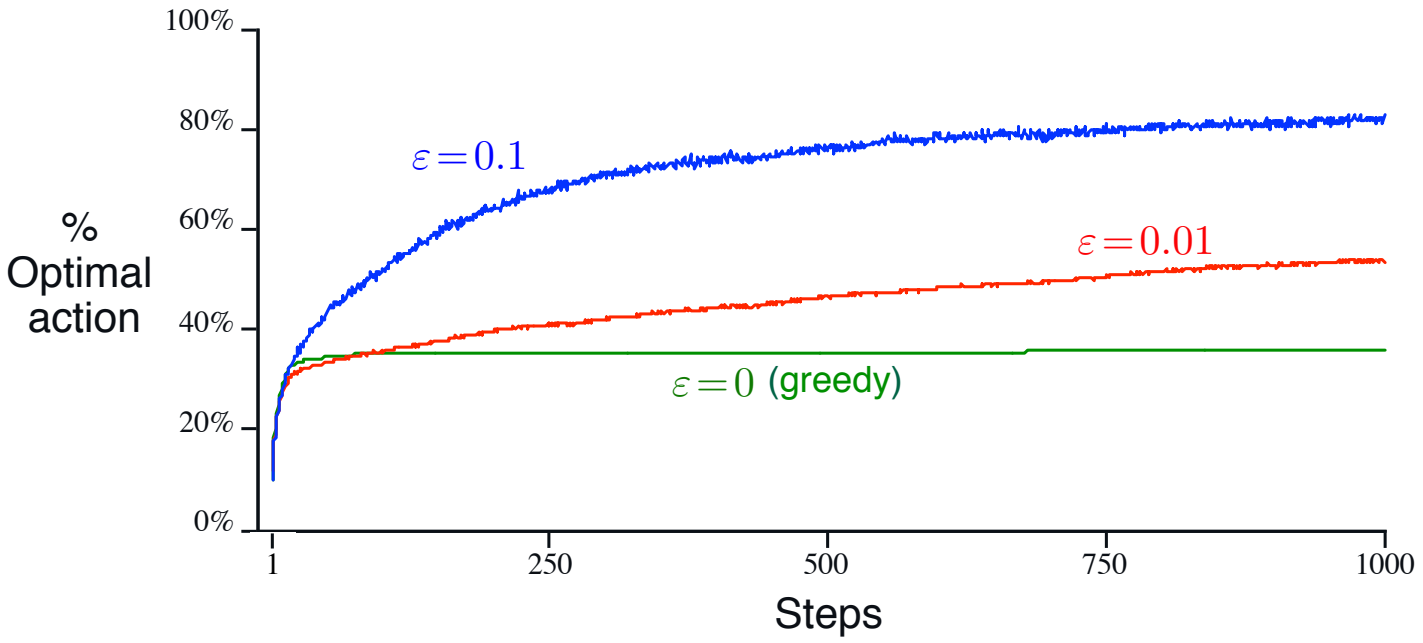


The agent explores 1% of the time

The agent performs only greedy steps

Performance ε-greedy

- Optimal action?

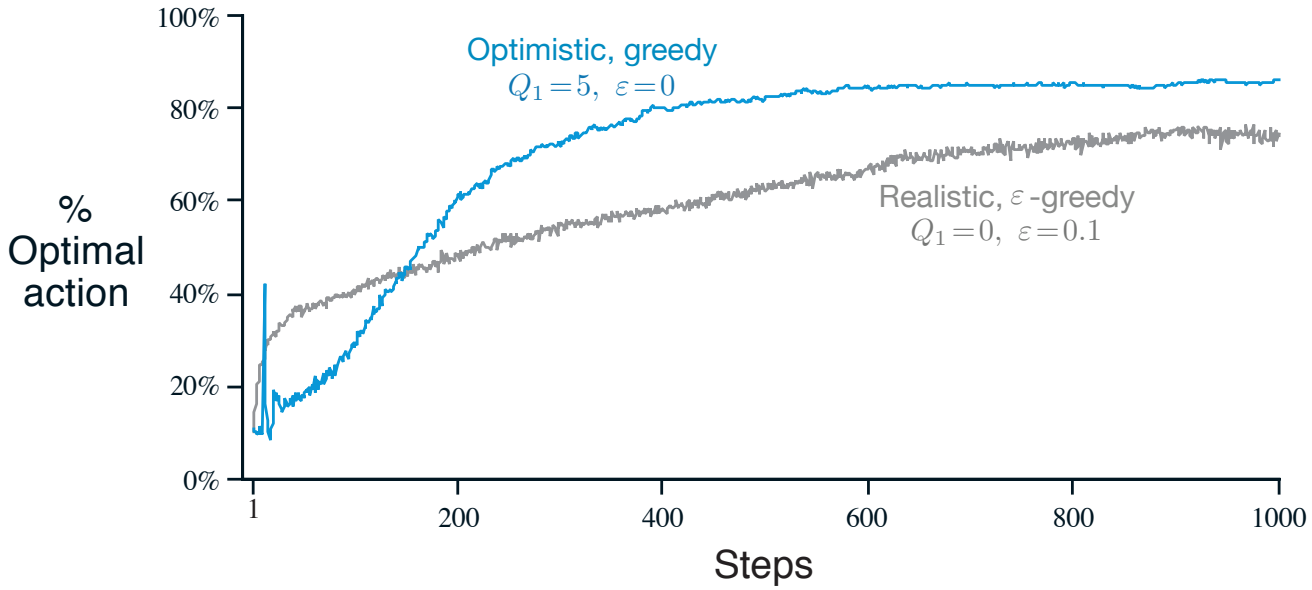


Experiments

- Let us run experiments for optimistic initial values method comparing
  - ε=0 (Greedy)
  - ε=0.1 (ε-greedy)

Performance of optimistic initial values

- Initial action values are used to encourage exploration
- In the 10-armed testbed we set all $q_1(a) = +5$, for all a
- All actions are tried several times before the value estimates converge
- The system does a fair amount of exploration even if greedy actions are selected all the time



Comments

- ε-greedy method
  - Explores ε% of the time
  - Depends on ε value
  - Depends on reward variance (small variance → less exploration to find the optimal action)
  - Suitable to nonstationary problems
- Optimistic initial values method
  - Encourages exploration
  - Is effective only stationary problems
  - Is far from being a generally useful approach to encouraging exploration
  - It is not well suited to nonstationary problems because it explores mainly at the beginning

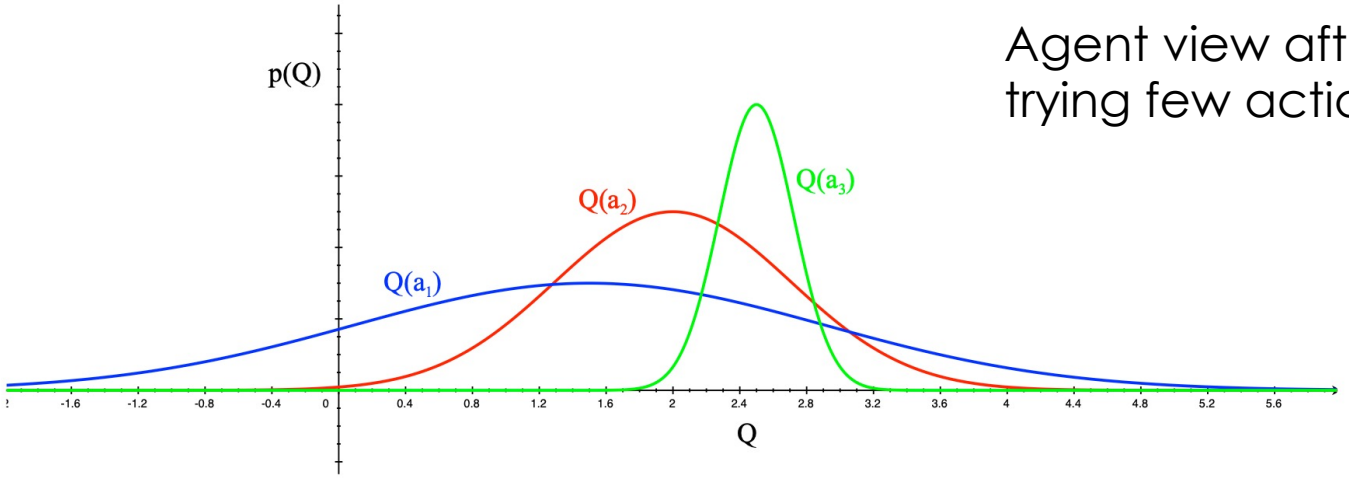
Approaches to action selection

- Naive Exploration
  - Add noise to greedy policy (e.g. ε-greedy)
- Optimistic Initialisation
  - Assume the best until proven otherwise
- Optimism in the Face of Uncertainty
  - Prefer actions with uncertain values

Uncertainty

- We have seen how to estimate action values from sampled rewards
- There is inherent **uncertainty** in the accuracy of our estimate
- Easy problem: two arms, one arm is always good, one arm is always bad, once you try both you are done (you always pick the best one)
- **Hard problem**: two arms, one arm is much better than the other one but there is much noise, and takes really long time to disambiguate (figure out that one arm is much better than the other one)
- Hard problems have similar-looking arms with different means

Optimism in the face of uncertainty

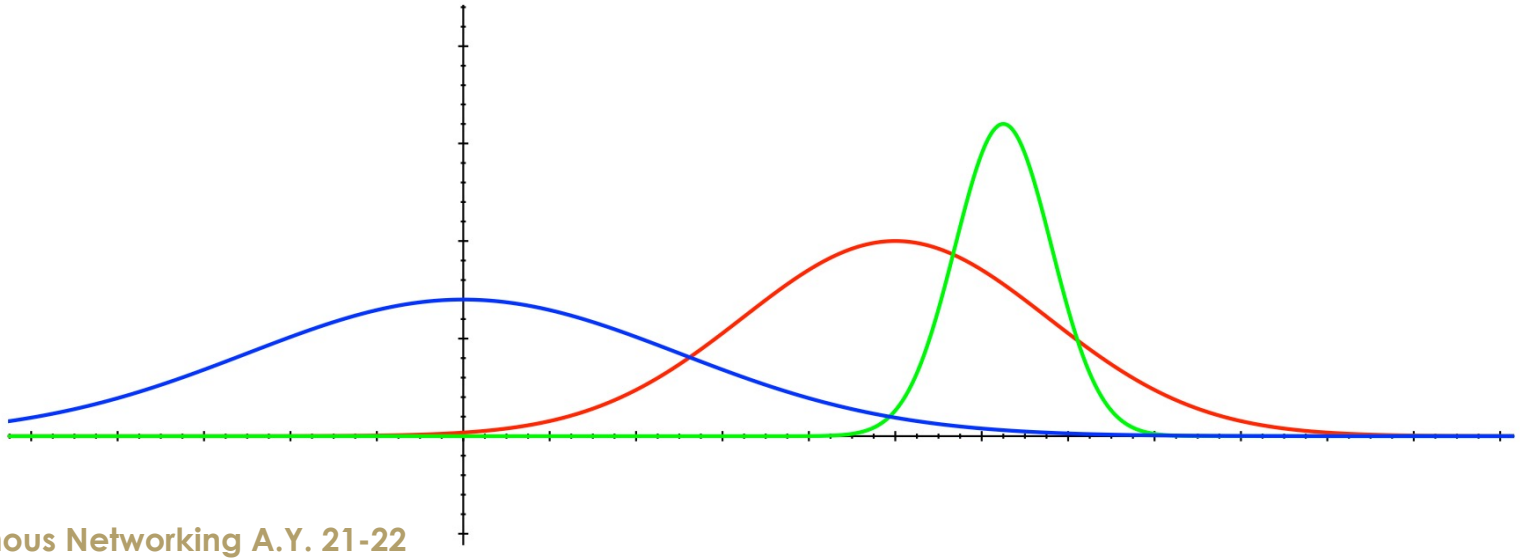


Agent view after trying few actions

- 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

- The optimism in the face of uncertainty principle says: ***do not take the arm you believe is best, take the one which has the most potential to be the best***
- 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



Optimism in the face of uncertainty

- So far we have seen how to estimate the mean but …
  - **How do we estimate uncertainty?**
  - **Can we reduce this uncertainty?**
- Then we can make better decisions (We are less uncertain about the values)

Uniform exploration

- ε-greedy

$$A_t \leftarrow \begin{cases} \underset{a}{\text{argmax}}\ Q_t(a) & \text{with probability } 1-\epsilon \\ a \sim Uniform(\{a_1 \ldots a_k\}) & \text{with probability } \epsilon \end{cases}$$

- Exploratory actions are selected uniformly
- Can we do better?

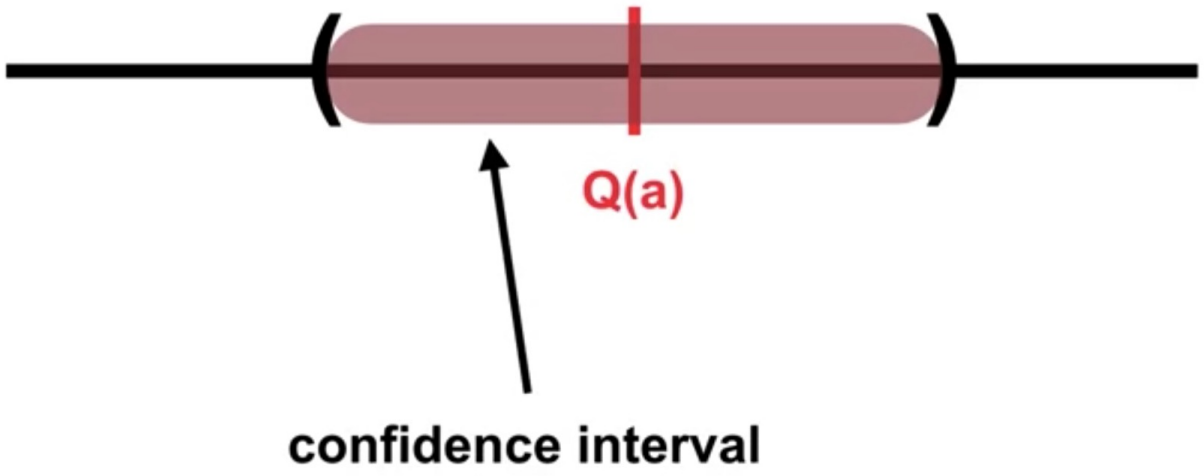
Uncertainty in estimates

- What does it mean to have uncertainty in the estimates?

- Q(a) represents our current estimate for action a.
- The brackets represent a confidence interval around q*(a)
- Brackets say we are **confident that the value of action a lies somewhere in this region**

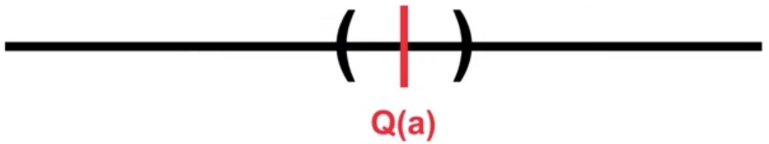
Uncertainty in estimates

- The region between the brackets is the confidence interval which represents our uncertainty.



Uncertainty in estimates

- If this region is very small, we are very certain that $q_*(a)$ is near our estimated value.



- If the region is large, we are uncertain that $q_*(a)$ is near our estimated value.

Upper Confidence Bound

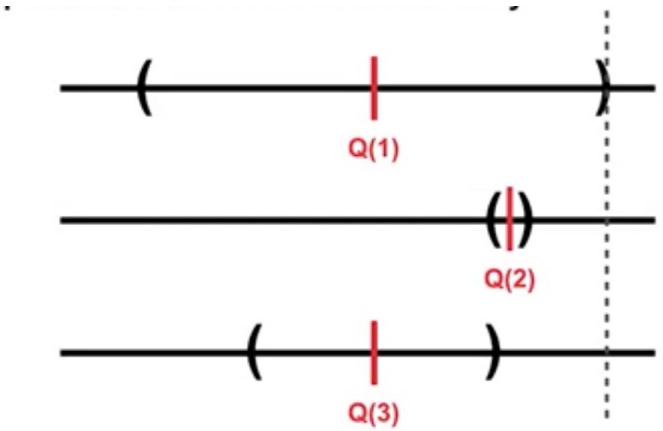
UCB follows the principle of optimism in the face of uncertainty

if we are **uncertain** about something, we should optimistically **assume that it is good**.

UCB: example

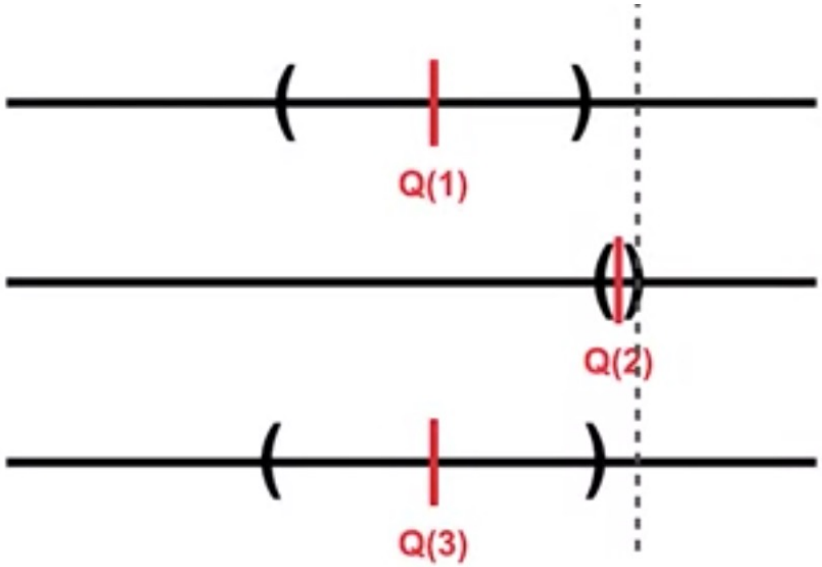
- We have three actions with associated uncertainties,
- Our agent has no idea which is best
- So it optimistically picks the action that has the highest upper bound
  - It does have the highest value and we get good reward

  OR
  - we get to learn about an action we know least about like the example on the slide.



UCB: example

- Let's let the algorithm pick one more action.
- This time Q2 has the highest upper-confidence bound because it's estimated value is highest, even though the interval is small



UCB action selection

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

exploitation

exploration

- We will select the action that has the **highest estimated value plus the upper-confidence bound exploration term**.
- The C parameter is a user-specified parameter that controls the amount of exploration

Example on exploration term

- 10.000 steps so far
- Imagine we have selected action a 5,000 times, then the uncertainty term here will be (0.043 * c)
- If instead we had only selected action a 100 times, the uncertainty term would be 10 times larger.

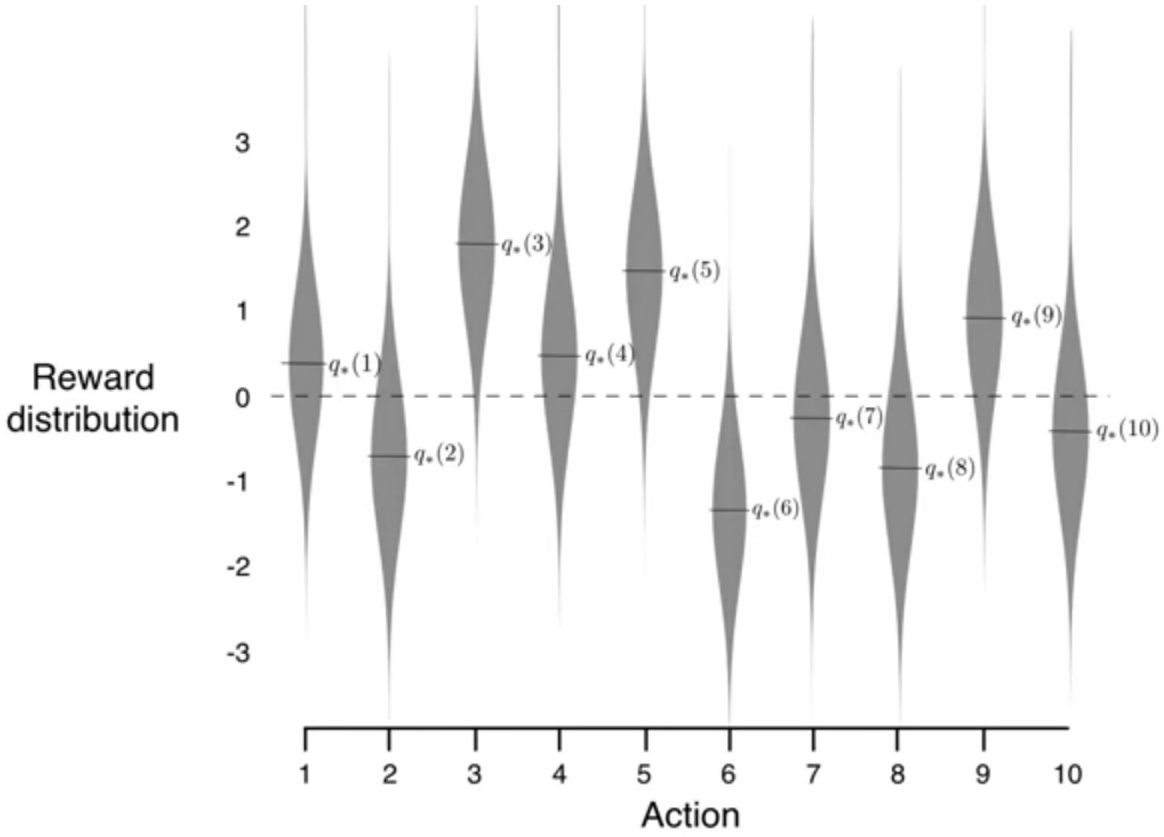
$$c\sqrt{\frac{\ln t}{N_t(a)}} \rightarrow c\sqrt{\frac{\ln \text{timesteps}}{\text{times action } a \text{ taken}}}$$

$$c\sqrt{\frac{\ln 10000}{5000}} \rightarrow 0.043c$$

$$c\sqrt{\frac{\ln 10000}{100}} \rightarrow 0.303c$$

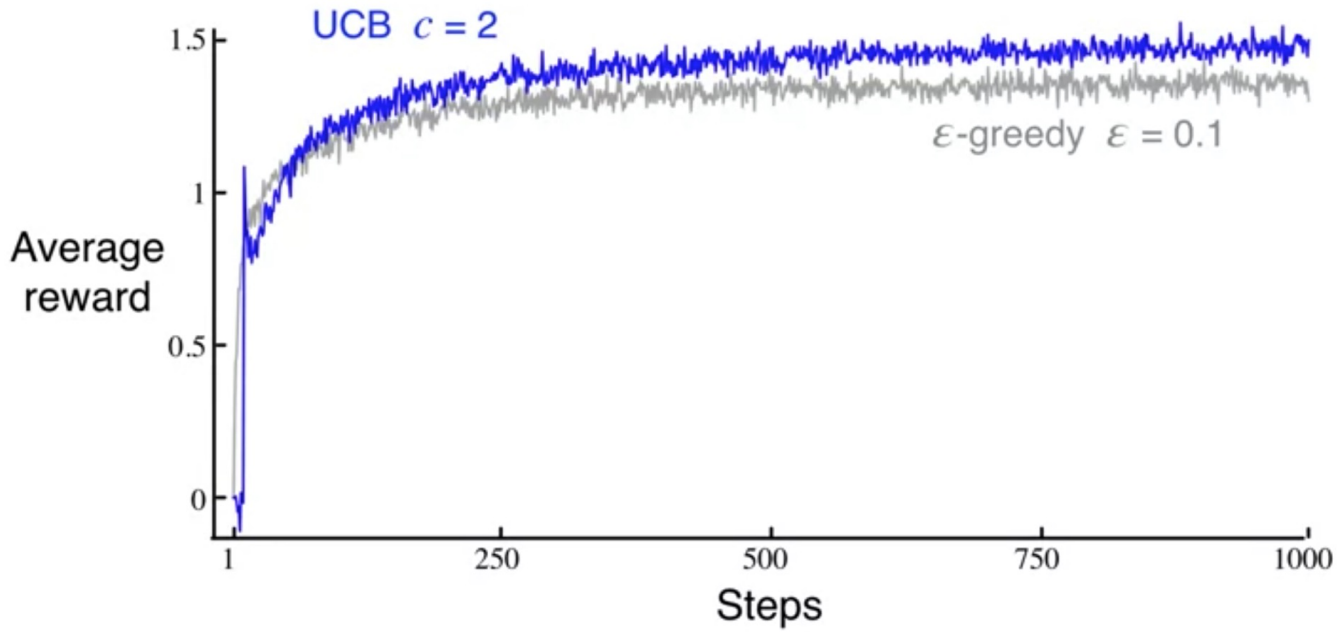
UCB performance

- The 10-armed testbed



UCB performance

- Performance of Upper Confidence Bound



- Initially, UCB explores more to systematically reduce uncertainty
- UCB's exploration reduces over time whereas Epsilon-greedy continues to take a random action 10 percent of the time

Conclusions

- Performance of strategies for action selection
  - Greedy
  - ε-greedy
  - Optimistic initial values
  - Upper Confidence Level
- UCB performs well but has difficulty in dealing with nonstationary problems