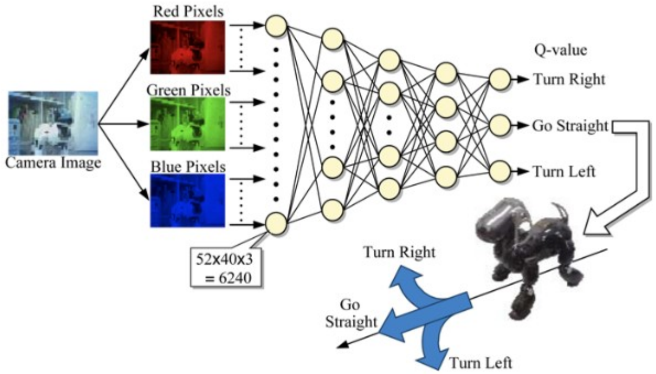
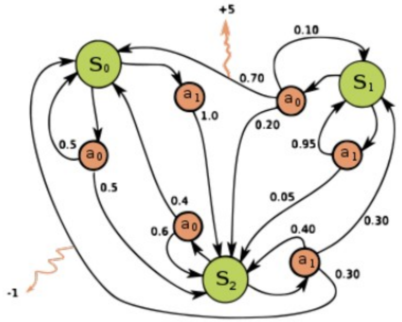
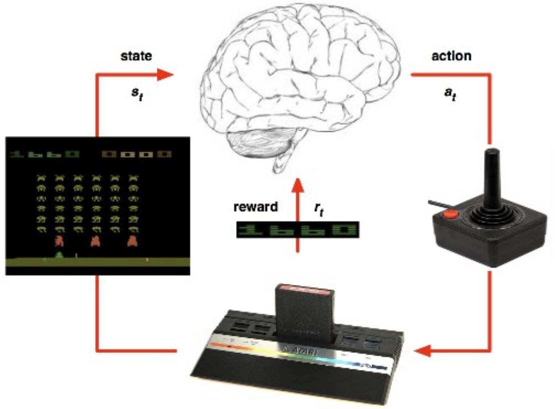
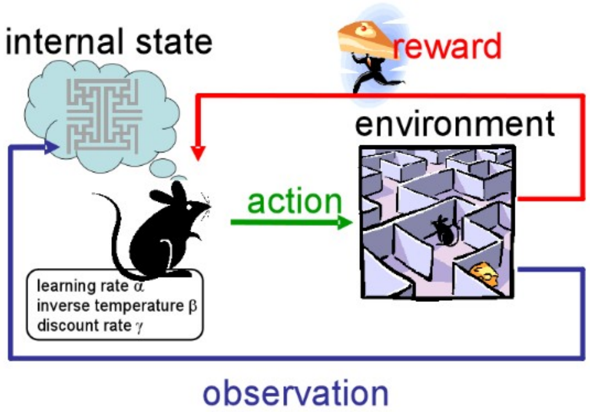
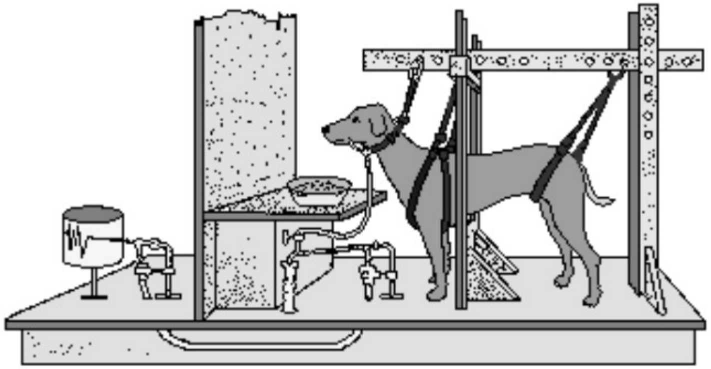
# COMP 138: Reinforcement Learning











# The Multi-Arm Bandit Problem

a.k.a. how to pick between Slot Machines (one-armed bandits) so that you walk out with the most $$$ from the casino

# How do we construct a value function at the start (before any actions have been taken)

| | | | |
|---|---|---|---|
| Zeros: | 0 | 0 | 0 |
| Random: | -0.23 | 0.76 | -0.9 |
| Optimistic: | +5 | +5 | +5 |

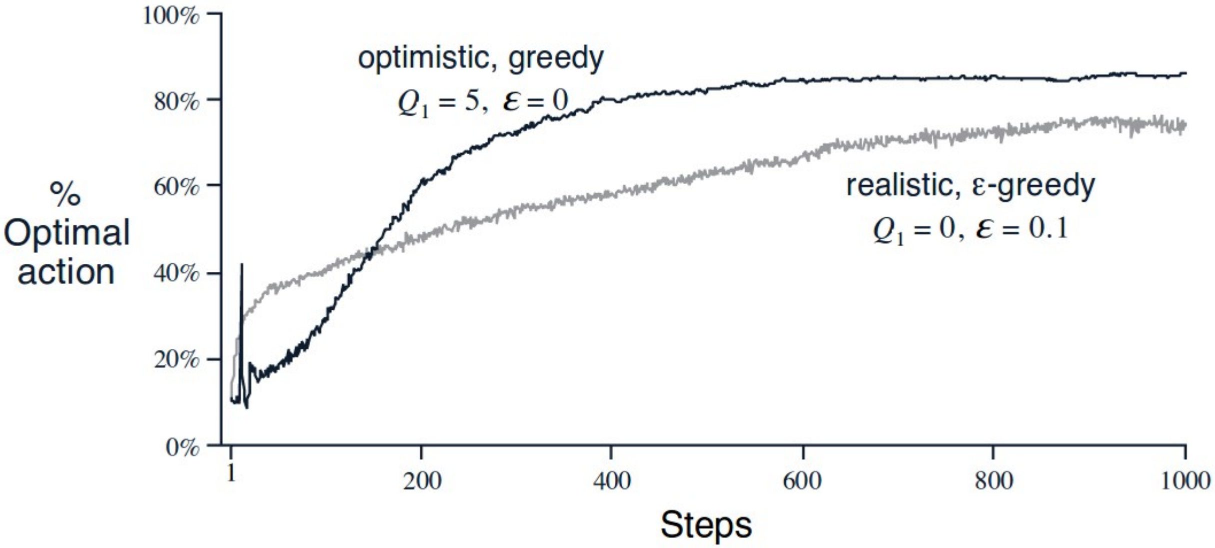
Arm 1

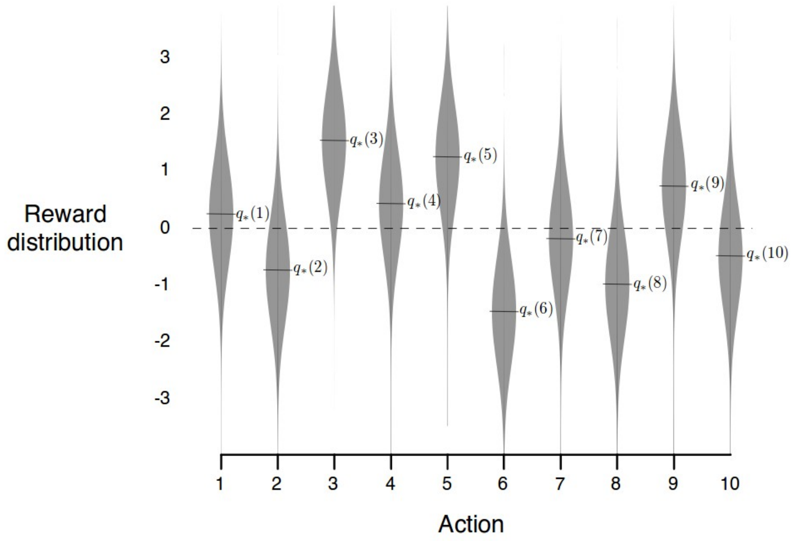
Arm 2

. . . .

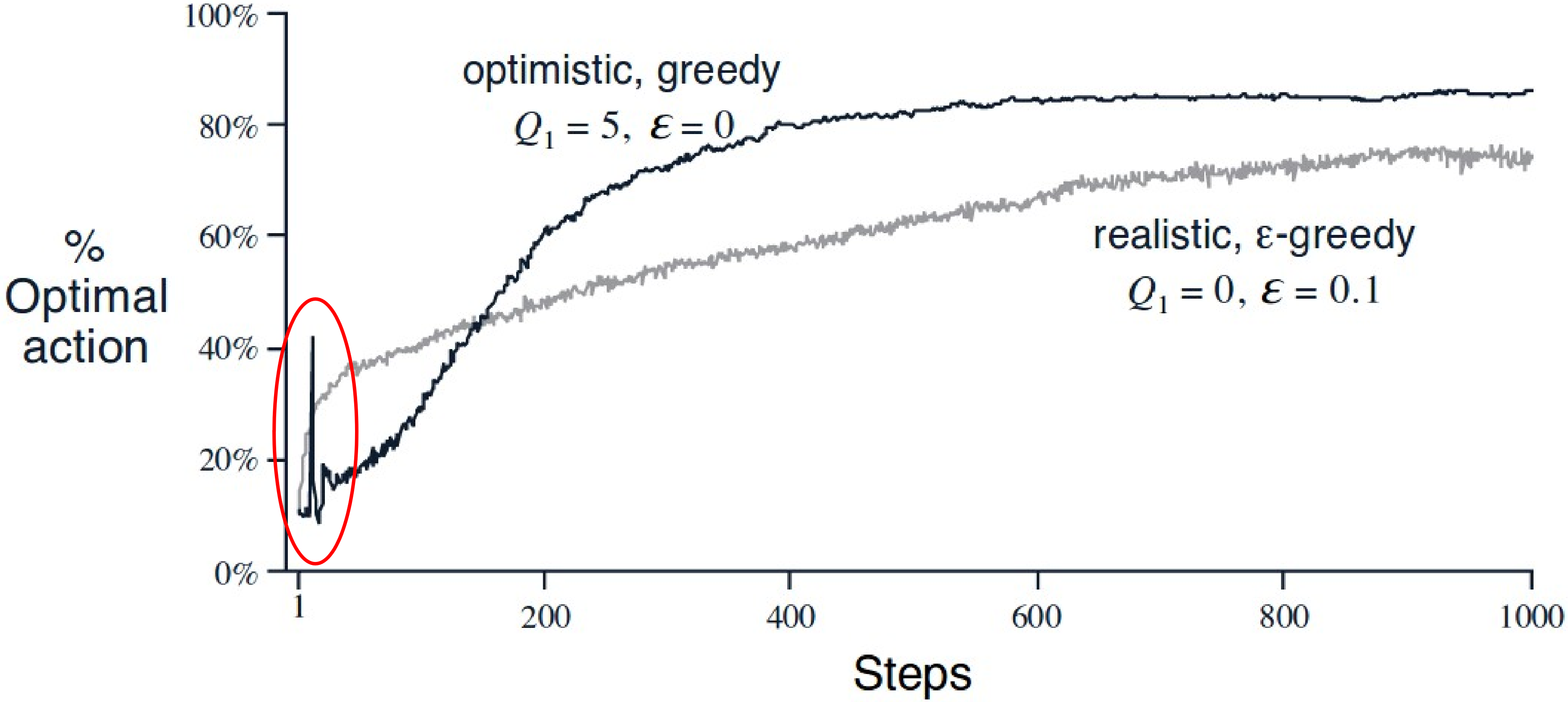
Arm k

# How do we construct a value function at the start (before any actions have been taken)





# Mysterious Spikes (Ex. 2.6)



# Exploration

- What’s wrong with epsilon-greedy exploration?

# Exploration

- What’s wrong with epsilon-greedy exploration?
- What are some ways we can explore in a more intelligent manner?

# Soft-Max Action Selection

Exponent of natural logarithm (~ 2.718)

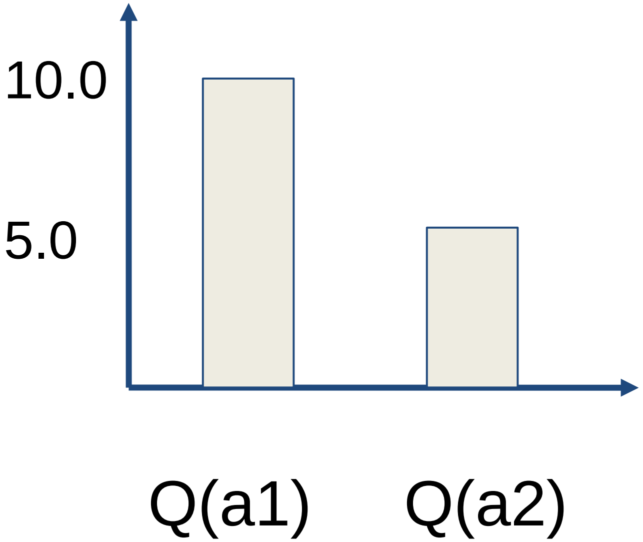
$$\frac{e^{Q_t(a)/\tau}}{\sum_{i=1}^{n} e^{Q_t(i)/\tau}}$$

“temperature”

As temperature goes up, all actions become nearly equally likely to be selected; as it goes down, those with higher value function outputs become more likely

# Example

Probability of picking a2 as a function of temperature:



(see graphing calculator)

$$\frac{e^{Q_t(a)/\tau}}{\sum_{i=1}^{n} e^{Q_t(i)/\tau}}$$

# Upper Confidence Bound Action Selection

# Upper Confidence Bound Action Selection

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

# Upper Confidence Bound Action Selection

Q-function at time t

natural logarithm of current time step

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

constant hyper-parameter which balances exploration vs exploitation

# of times action *a* has been picked

# Upper Confidence Bound Action Selection

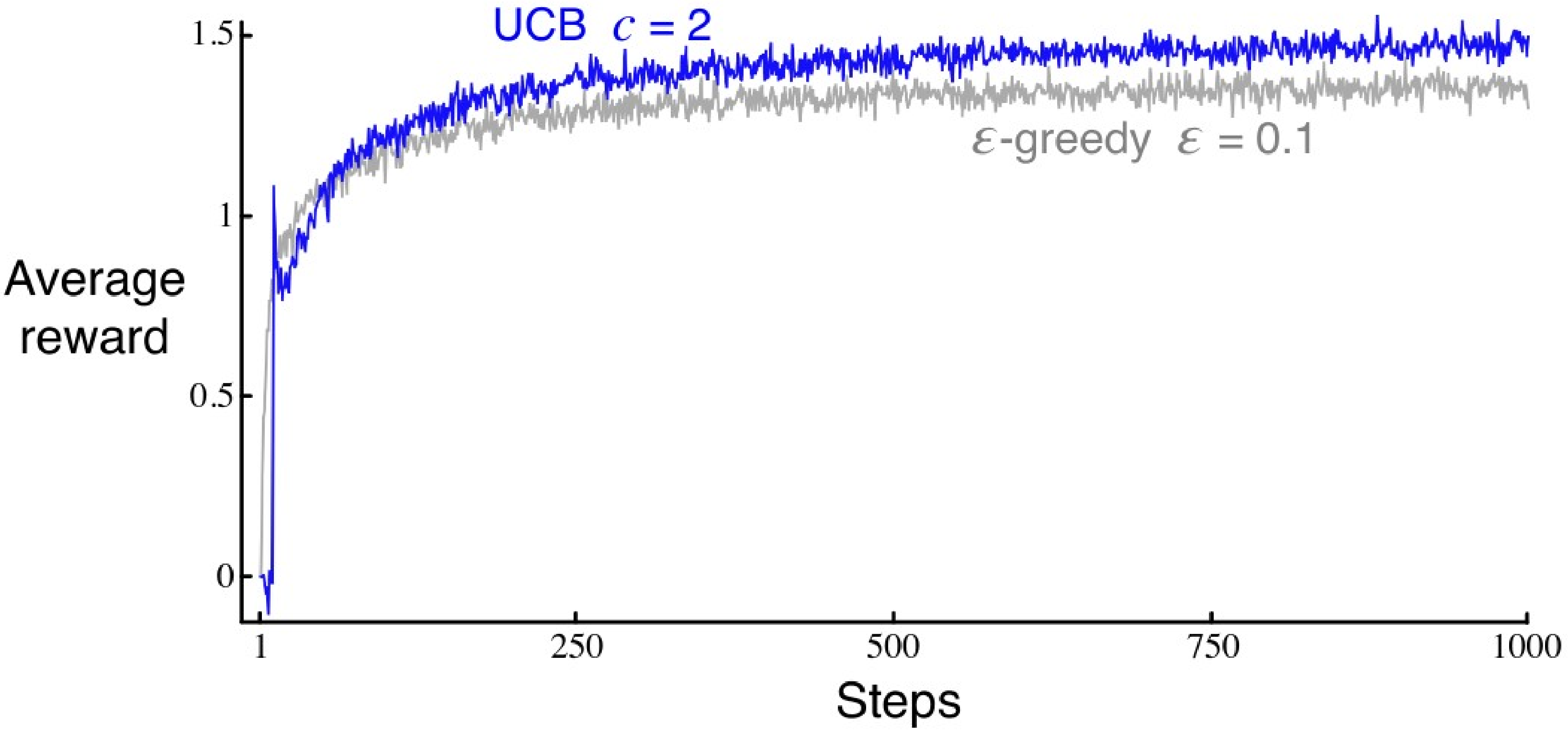


**Figure 2.4:** Average performance of UCB action selection on the 10-armed testbed. As shown, UCB generally performs better than $\varepsilon$-greedy action selection, except in the first $k$ steps, when it selects randomly among the as-yet-untried actions.

# Upper Confidence Bound Action Selection

UCB Spikes In Figure 2.4 the UCB algorithm shows a distinct spike in performance on the 11th step. Why is this? Note that for your answer to be fully satisfactory it must explain both why the reward increases on the 11th step and why it decreases on the subsequent steps. Hint: if c=1, then the spike is less prominent.

# So…which method is best?

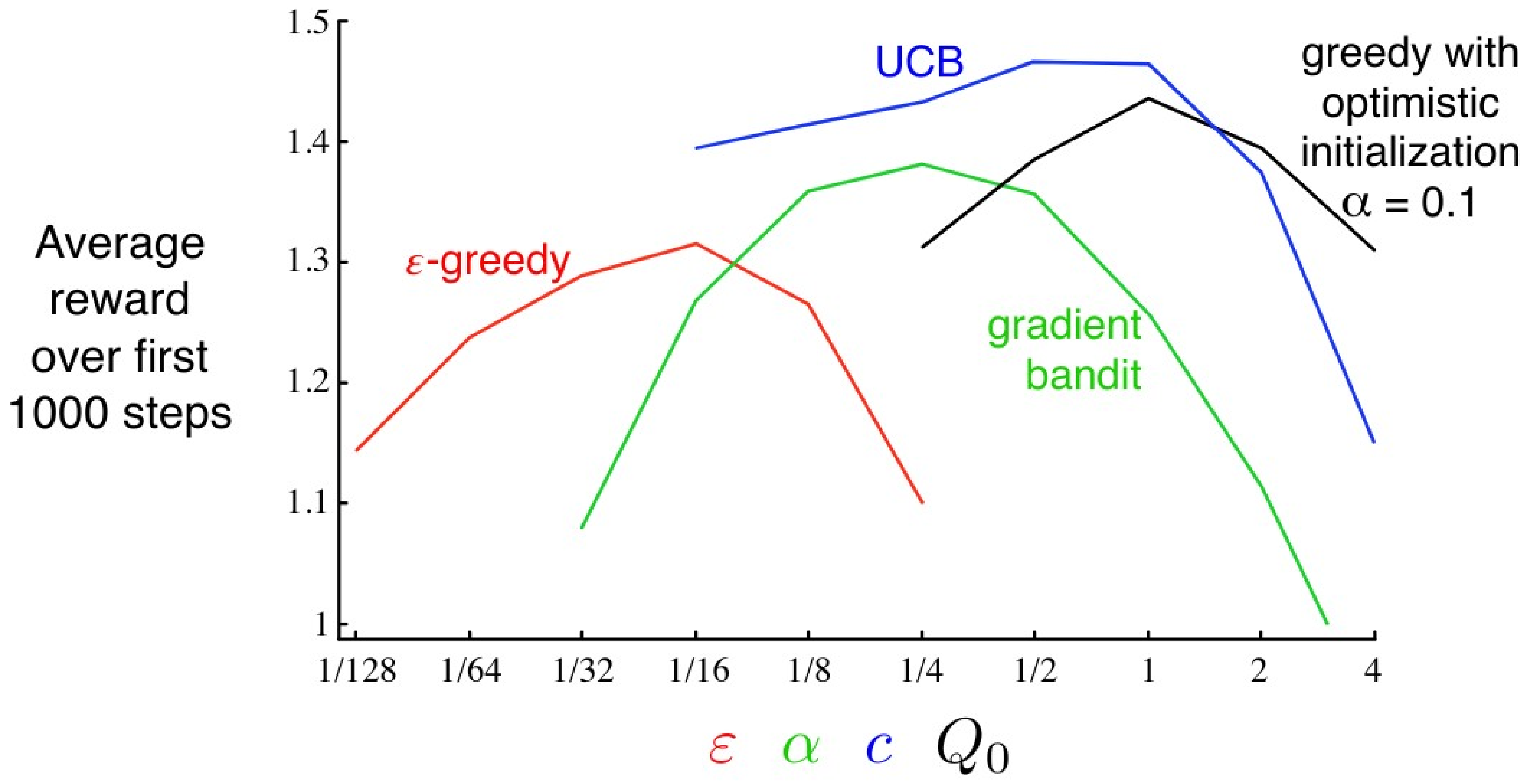


**Figure 2.6:** A parameter study of the various bandit algorithms presented in this chapter. Each point is the average reward obtained over 1000 steps with a particular algorithm at a particular setting of its parameter.

# Further Reading

- Bandit problems and the exploration/exploitation tradeoff
- https://dl.acm.org/doi/abs/10.1109/4235.728210

# Reading Responses

“My question is: Is the Epsilon value selection/decay rate selection an art or a science? For certain action space and state space sizes, are there ranges of epsilon values that are most optimal for convergence? Or does this not matter?”

- Cameron

# Reading Responses

“Why is it necessary when taking a greedy action to choose randomly between all max value actions? How would using a deterministic method to choose the max value action distort an algorithm’s results?”

– Tyler

# Reading Responses

“The balance between trying new things and sticking to what we know, like in the multi-armed bandit problem, seems pretty relevant. For example, how do big companies, such as banks, or platforms like Netflix, use these concepts? And the whole idea of Associative Search or Contextual Bandits got me thinking. With so much data around us, it's crucial to figure out how actions tie back to specific scenarios. For recommendation systems, for instance. How do they balance between suggesting familiar content and throwing in something new?”

– Yinkai

# Reading Responses

“One big confusion that I had after the chapter was in regard to the optimistic initial value greedy method ... Wouldn't the agent just try each action once, and then pick the action that had the best first reward for the rest of the problem? How does it perform so well compared to the seemingly "smarter" techniques?”

– Andrew

# Reading Responses

“Question: Is it possible to ensure we have an optimal policy supposing that the environment is *stationary enough*? For instance, if we can precisely quantify how a given environment might change, can we tune an $\alpha$ in real time to stay converging to optimal, or bound our error from optimal?”

- Jacob

# Reading Responses

“For chapter 2, when it comes to Upper Confidence Bounds, why is the numerator term in the square root "ln t". Does the natural log provide specific value, or is just serving as a slow increment? Could it be replaced with anything?”

– Hayley

# Reading Responses

“In class we discussed the idea of a decaying Ɛ where the frequency of random action decreases over time. In theory, this mirrors a parallel decrease in uncertainty. What if uncertainty can also increase? Are there algorithms where Ɛ instead tracks the variability of rewards (so Ɛ increases if a highly unexpected reward is seen)?”

– Tyler

# Reading Responses

“Plots of the three method performances are presented and UCB seems to behave a lot better. Can we just draw the conclusion that UCB is the most powerful? Or any other aspects should be taken into consideration as well?”

– Linda

# Reading Responses

“… failures seem unavoidable while the agents explore new actions. Chances are that they may reach extremely dangerous situations, like breaking an arm of a robot. Are there research working on addressing such safety concerns while exploring new actions? If so, do they typically set a very small negative reward signal upon reaching dangerous states, or do they introduce new signals to manually stop them from entering such states beforehand?”

– Wenchang

# Reading Responses

“Thinking about the Tic-tac-toe game, is there a certain amount of “rules” that the system should know before hand? Or can there be an element of also learning the rules of the game as it goes and if so, how much does the system need to know before it can “play””

– Kelly

# Reading Responses

“Question1: While reading, I found the mathematical equations in Chapter 2, such as equation 2.12 in the gradient bandit algorithm section, hard to follow. Should we master the mathematical equations in the book, or is understanding the general idea sufficient, for example, understand how each parameter affects the results?”

- Song

# MAB TF library introduction

https://www.youtube.com/watch?v=7QFSziiAnxI

# MAB Research

- Identifying outlier arms in MAB problems (NeurIPS 2017): https://www.youtube.com/watch?v=ecgNELgTzS8
- Finding Structure in MABs: https://www.youtube.com/watch?v=Thmh_--kVmg
- Budgeted Combinatorial MABs (AAMAS 2022): https://www.youtube.com/watch?v=gEigPlsmJ0M
- Contextual Bandits in Healthcare: https://www.youtube.com/watch?v=gbZHmPJaU0I

# Looking Deeper

- Contextual bandits
- Combination Actions
- Continuous Actions
- Continuous Contexts + Continuous Actions
- Discrete but parameterized actions
- ...

# ChatGPT Policy