“And always, he fought the temptation to choose a clear, safe course, warning “That path leads ever down into stagnation.””

Frank Herbert, *Dune*

CMPUT 365
Introduction to Sequential-Decision Making

Marlos C. Machado

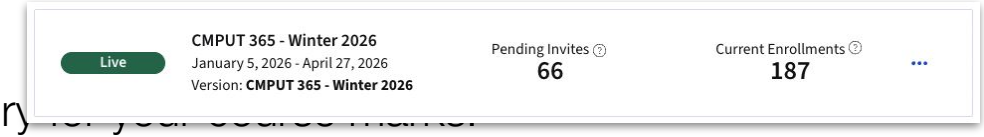
Classes 3 & 4 of 35

Plan

- Motivation
- *Non-comprehensive* overview of Intro to Sequential-Decision Making in Coursera (Bandits, Chapter 2 of the textbook)

Reminder

You **should be enrolled in the private session** we created in Coursera for CMPUT 365.



I **cannot** use marks from the public repository for your course marks.

You **need** to **check**, **every time**, if you are in the private session and if you are submitting quizzes and assignments to the private section.

The deadlines in the public session **do not align** with the deadlines in Coursera.

If you have any questions or concerns, **talk with the TAs** or email us `cmput365@ualberta.ca`.

Notes

- Office hours start next week:
  - Lucas Monday 16:00 – 18:00 @ UCOMM 2-138
  - Yuyang Tuesday 13:00 – 15:00 @ UCOMM 2-138
  - Shashank Wednesday 10:30 – 12:30 @ UCOMM 2-138
  - Aaron Wednesday 14:00 – 16:00 @ UCOMM 2-138
  - Siddarth Thursday 11:00 – 13:00 @ UCOMM 3-138
  - Dasha Thursday 13:00 – 15:00 @ UCOMM 2-138
  - Diego Friday 10:00 – 12:00 @ UCOMM 3-138
  - Parham Friday 15:00 – 17:00 @ UCOMM 2-138

Please, interrupt me at any time!

https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Let’s play a game!

https://m.media-amazon.com/images/I/91q+3X0sUZL._AC_SL1500_.jp

Bandits

Arm 1	Arm 2	Arm 3

Reinforcement learning (RL)

- RL is about learning from *evaluative* feedback (an evaluation of the taken actions) rather than *instructive* feedback (being given the correct actions).
  - Exploration is essential in reinforcement learning.

- It is not necessarily about online learning, as said in the videos, but more generally about sequential decision-making.

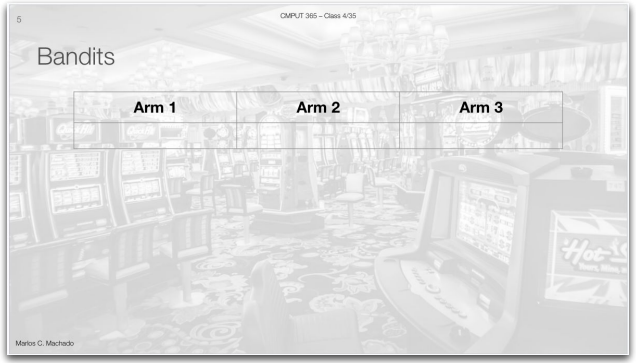
- Reinforcement learning potentially allows for continual learning but in practice, quite often we deploy our systems.

Why study bandits?

- Bandits are the simplest possible reinforcement learning problem.
  - Actions have no delayed consequences.

- Bandits are deployed in so many places! [Source: Csaba's slides]
  - Recommender systems (Microsoft paper):
    - News,
    - Videos,
    - …
  - Targeted COVID-19 border testing (Deployed in Greece, paper).
  - Adapting audits (Being deployed at IRS in the USA, paper).
  - Customer support bots (Microsoft paper).
  - … and more.

Why study bandits?

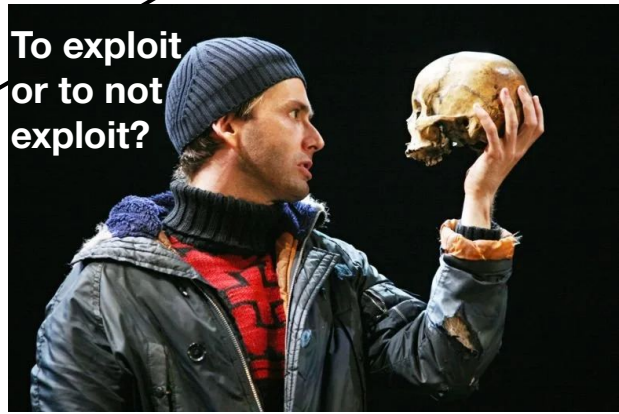


We don't really know q*, so we use an estimate of it, Q_t

$$q^*(a) \doteq \mathbb{E}[R_t \mid A_t = a]$$

$$A_t \doteq \text{argmax}_a \, Q_t(a)$$

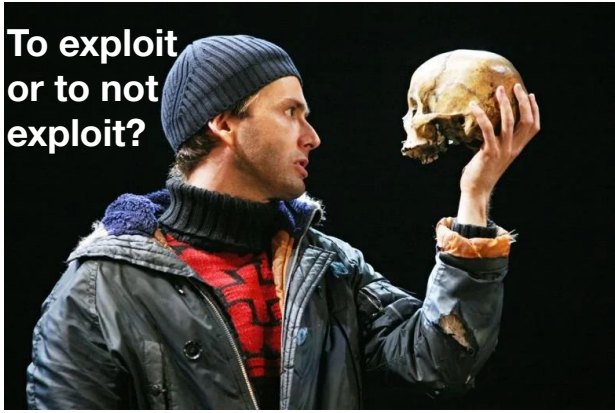
Greedy action



https://shakespeareanstudent.files.wordpress.com/2019/10/fb89ff1a-6f63-11e6-acba-85f5c900fc1a1735507721165554264.

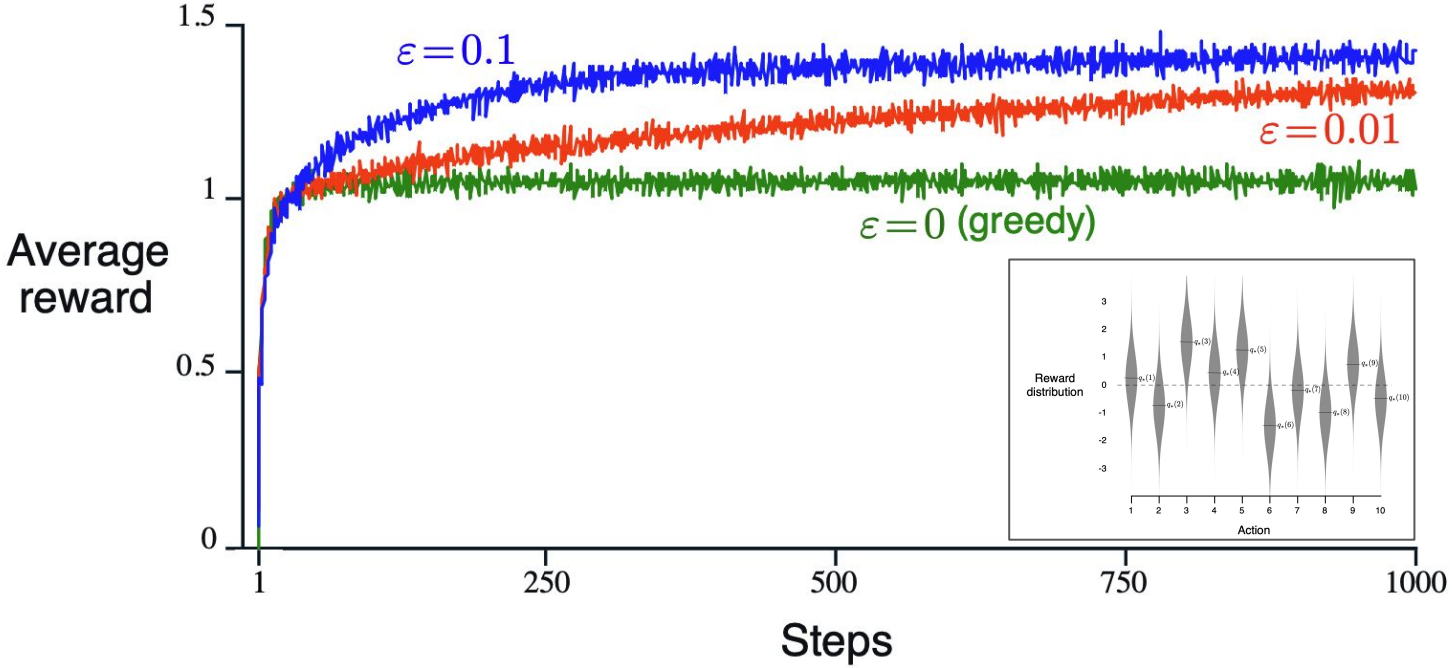
Exploration

- Exploration is the opposite of exploitation.
- It is a whole, very active area of research, despite the textbook not focusing on it.
- How can we explore?
  - Randomly (ε-greedy)
  - Optimism in the face of uncertainty
  - Uncertainty
  - Novelty / Boredom / Surprise
  - Temporally-extended exploration
  - …



https://shakespeareanstudent.files.wordpress.com/2019/10/fb89ff1a-6f63-11e6-acba-85f5c900fc1a1735507721165554264.

Exploration matters



Incremental updates to estimate q_*

$$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i$$

Incremental updates to estimate q_*

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

Update rule

NewEstimate ← OldEstimate + StepSize [Target - OldEstimate]

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

A bigger step-size means bigger steps (updates).

A constant step-size gives more weight to recent rewards.

How you initialize Q_n really matters.

The principle of **optimism in the face of uncertainty** really leverages that.

This is the direction you need to move to get closer to the solution.

A note on step-sizes

A well-known result in stochastic approximation theory gives us the conditions required to assure convergence with probability 1:

$$\sum_{n=1}^{\infty} \alpha_n(a) = \infty$$

and

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

Cannot be too small.
E.g.: $\alpha_n = 1/n^2$

Cannot be too big.
E.g.: $\alpha_n = 1$

A constant step-size is biased

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

A constant step-size is biased

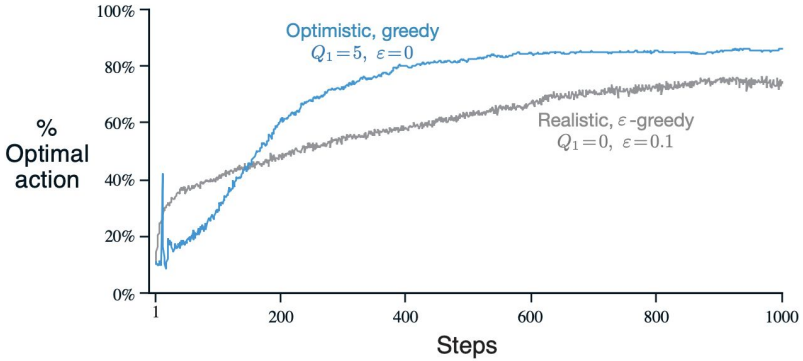
$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha)Q_n \\
&= \alpha R_n + (1-\alpha)\left[\alpha R_{n-1} + (1-\alpha)Q_{n-1}\right] \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1} \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 \alpha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

Q_1 is always there, forever, impacting the final estimate.

Optimism in the face of uncertainty

$$Q_{n+1} = Q_n + \alpha\left[R_n - Q_n\right]$$

Idea: Initialize Q_0 to an overestimation of its true value (optimistically).

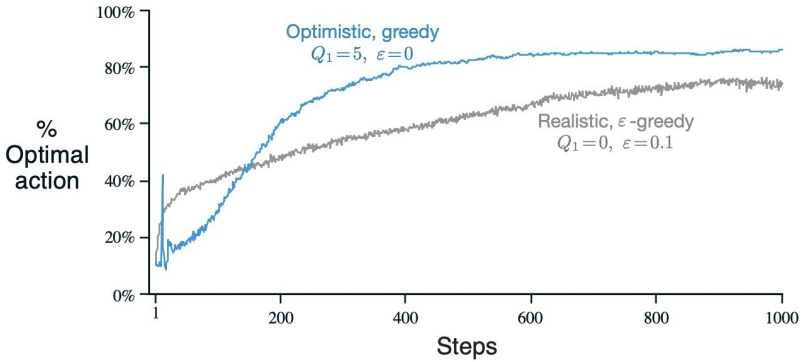


Optimism in the face of uncertainty

$$Q_{n+1} = Q_n + \alpha\left[R_n - Q_n\right]$$

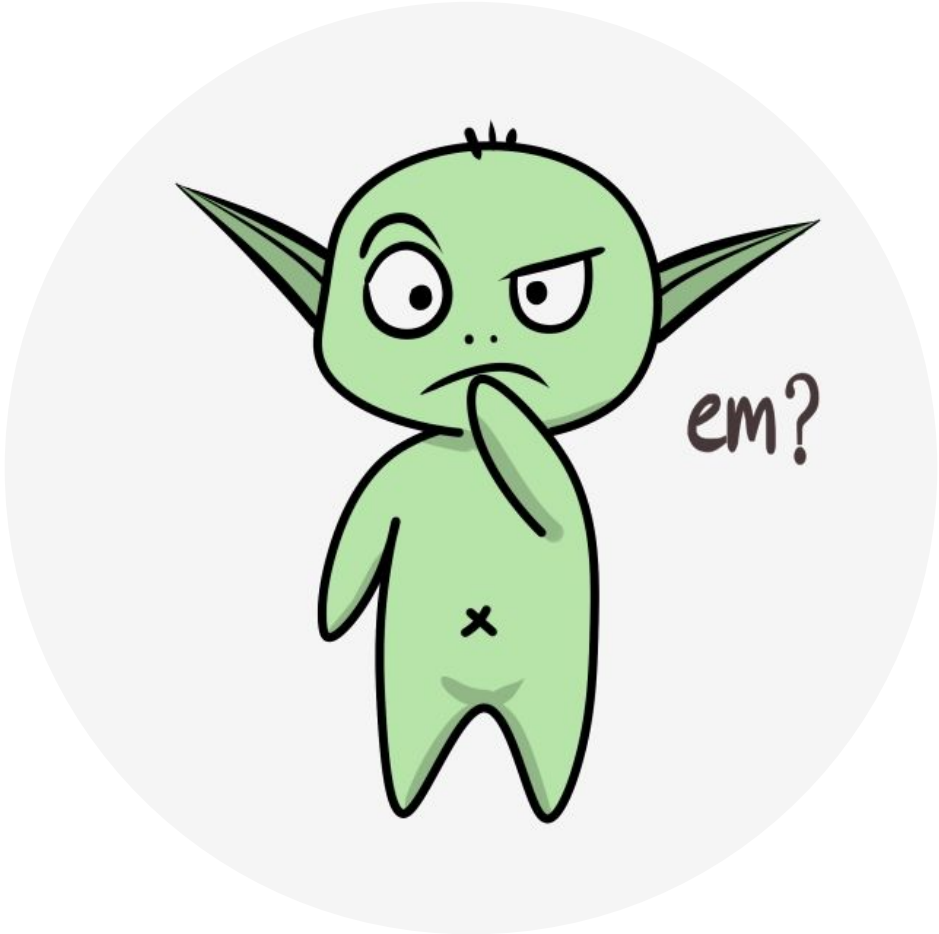
Idea: Initialize Q_0 to an overestimation of its true value (optimistically).

- You either maximize reward or you learn from it.
- The value you initialize Q_0 can be seen as a hyperparameter and it matters.
- There are equivalent transformations in the reward signal to get the same effect.
- For bandits, UCB uses an upper confidence bound that with high probability is an overestimate of the unknown value.



How do we choose the best hyperparameter (α, ε, c, etc)?

- For this course: we try many things out and see what works best ¯_(ツ)_/¯



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Upper-Confidence-Bound Action Selection

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

Theorem 1. *For all $K > 1$, if policy* UCB1 *is run on K machines having arbitrary reward distributions $P_1, \ldots, P_K$ with support in $[0, 1]$, then its expected regret after any number n of plays is at most*

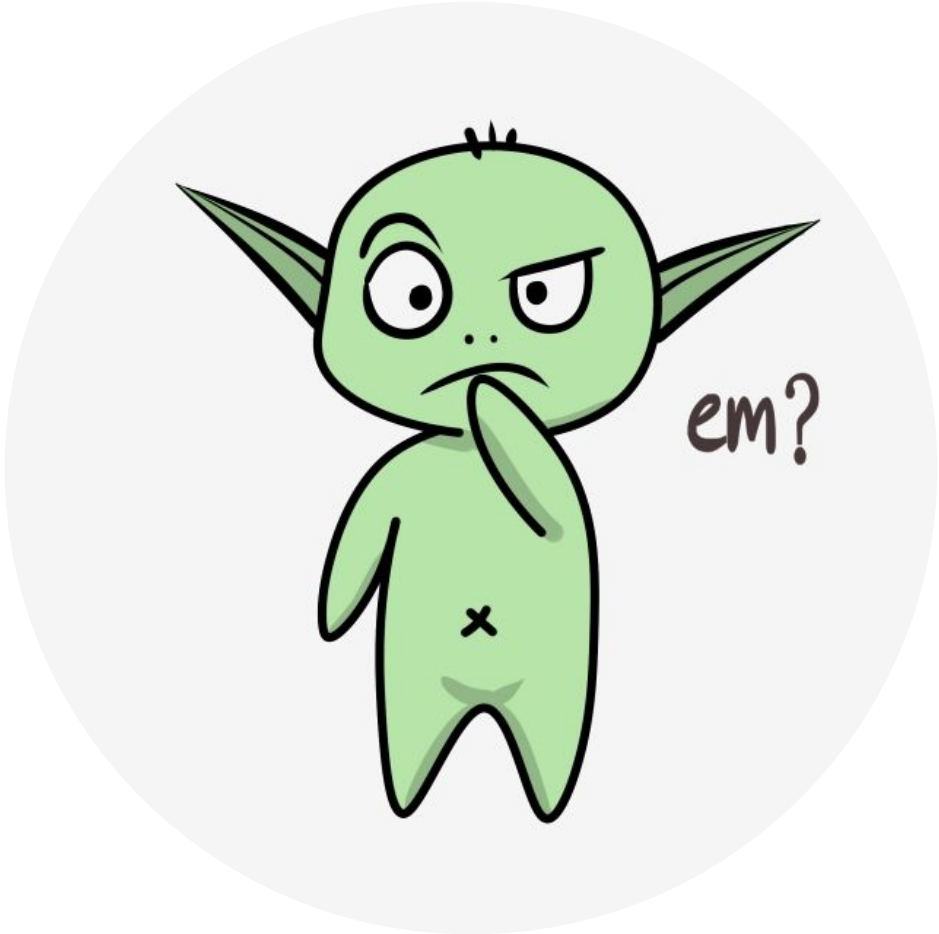
$$\left[8 \sum_{i:\mu_i < \mu^*} \left(\frac{\ln n}{\Delta_i} \right) \right] + \left(1 + \frac{\pi^2}{3} \right) \left(\sum_{j=1}^{K} \Delta_j \right)$$

where $\mu_1, \ldots, \mu_K$ are the expected values of $P_1, \ldots, P_K$.

Auer, Cesa-Bianchi, and Fischer (2002), *Machine Learning*.

Contextual bandits (Associative search)

- One need to associate difference actions with different *situations*.
- You need to learn a *policy,* which is a function that maps situations to actions.
- Most real-world problems modeled as bandits problems are modeled as contextual bandits problems.
- Example: A recommendation system, which is obviously conditioned on the user to which the system is making recommendations to.



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Question 1. Suppose a game where you choose to flip one of two (possibly unfair) coins. You win \$1 if your chosen coin shows heads and lose \$1 if it shows tails.

(a) Model this as a K-armed bandit problem: define the action set.

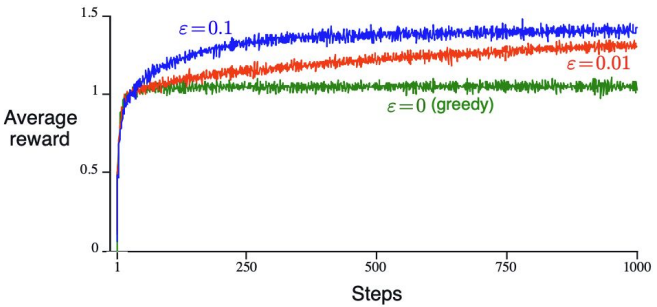
(b) Is the reward a deterministic or stochastic function of your action?

(c) You do not know the coin flip probabilities. Instead, you are able to view 6 sample flips for each coin respectively: (T,H,H,T,T,T) and (H,T,H,H,H,T). Use the sample average formula (equation 2.1 in the book) to compute the estimates of the value of each action.

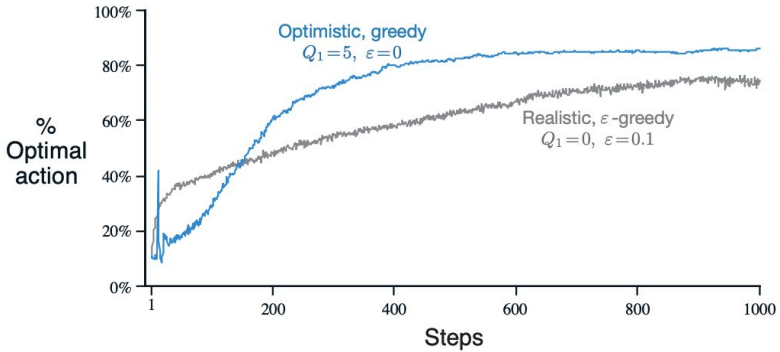
(d) Decide on which coin to flip next! Assume it's an exploit step.

Exercise 2.2: Bandit example Consider a k-armed bandit problem with $k = 4$ actions, denoted 1, 2, 3, and 4. Consider applying to this problem a bandit algorithm using ε-greedy action selection, sample-average action-value estimates, and initial estimates of $Q_1(a) = 0$, for all a. Suppose the initial sequence of actions and rewards is $A_1 = 1$, $R_1 = -1$, $A_2 = 2$, $R_2 = 1$, $A_3 = 2$, $R_3 = -2$, $A_4 = 2$, $R_4 = 2$, $A_5 = 3$, $R_5 = 0$. On some of these time steps the ε case may have occurred, causing an action to be selected at random. On which time steps did this definitely occur? On which time steps could this possibly have occurred? □

Exercise 2.3 In the comparison shown in Figure 2.2, which method will perform best in the long run in terms of cumulative reward and probability of selecting the best action? How much better will it be? Express your answer quantitatively. □



Exercise 2.6: Mysterious Spikes The results shown in Figure 2.3 should be quite reliable because they are averages over 2000 individual, randomly chosen 10-armed bandit tasks. Why, then, are there oscillations and spikes in the early part of the curve for the optimistic method? In other words, what might make this method perform particularly better or worse, on average, on particular early steps? □



Exercise 2.7: Unbiased Constant-Step-Size Trick In most of this chapter we have used sample averages to estimate action values because sample averages do not produce the initial bias that constant step sizes do (see the analysis leading to (2.6)). However, sample averages are not a completely satisfactory solution because they may perform poorly on nonstationary problems. Is it possible to avoid the bias of constant step sizes while retaining their advantages on nonstationary problems? One way is to use a step size of

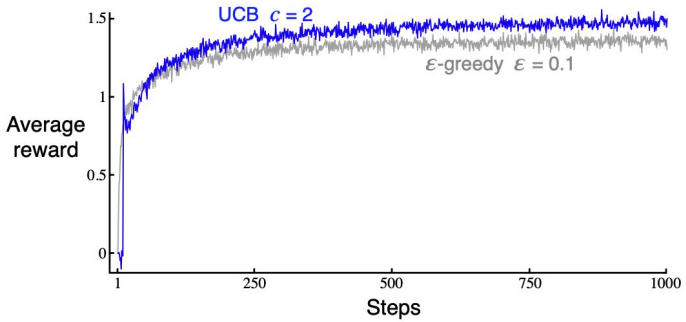
$$\beta_n \doteq \alpha/\bar{o}_n, \tag{2.8}$$

to process the nth reward for a particular action, where $\alpha > 0$ is a conventional constant step size, and $\bar{o}_n$ is a trace of one that starts at 0:

$$\bar{o}_n \doteq \bar{o}_{n-1} + \alpha(1 - \bar{o}_{n-1}), \quad \text{for } n > 0, \quad \text{with } \bar{o}_0 \doteq 0. \tag{2.9}$$

Carry out an analysis like that in (2.6) to show that Q_n is an exponential recency-weighted average *without initial bias*. □

Exercise 2.8: 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. □



Next class

Reminder: Practice Quiz and Programming Assignment for Coursera's Fundamentals of RL: Sequential decision-making is due on Monday.

- What **I** plan to do on Monday: Wrap up Fundamentals of RL: An introduction to sequential decision-making (Bandits)
  - Time permitting, we'll work on some exercises in the classroom.